THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

Test and Validation of the Integrity and Performance of High Speed Interfaces

Student: James Gabauer
Student Number: 43132672
Course: ENGG7290
Academic Supervisor: Matt D'Souza
Industry Supervisor: Ken Wilson
Placement Institution: Opengear
Submission Date: 27 June 2019

Faculty of Engineering, Architecture and Information Technology

A thesis submitted in partial fulfilment of the requirements of the Bachelor of Engineering/Master of Engineering

Acknowledgments

There are a number of people who have contributed to not just this project but my whole academic career, by supporting and advising me. First, I would like to start by thanking my academic supervisor Matt D'Souza for his help over the years, but more specifically this semester. I would also like to thank the team at Opengear, specifically Ken Wilson, Tim Gibson, Ash Boldaji, Jordan Conley, and Tony Merenda. You all provided advice and feedback, without which, I would probably still be trying to scope the project or I would be stuck debugging code I had written. I would like to thank the whole Gabauer family, and especially my Mum and Dad for their support over the course of my academic career, and my whole life really, who have tolerated the various degree changes I have been through to get here. I would like to thank all of my friends from UQ but especially my boy, Reuben Madden, we did it man, top of the sinking ship! Last but not least I would like to thank my partner Lizzie her patience, support, and cooking have kept me alive to make this possible. Thank you.

Executive Summary

High speed interfaces have become commonplace in modern computing for internal and external peripheral connections. As demand for high transfer rates increases, it becomes increasingly important that these interfaces can be tested and validated during development. This can be done through the purchase of specialised test equipment and software, or through an external provider. Typically these solutions provide full interface compliance testing, and are costly and often unnecessary during prototype development. As a designer and manufacturer of console servers, Opengear utilises a number of high speed interfaces across their range of products. The larger product iterations revolve around a main board with different add-on boards connected via a PCI-E interface to distinguish different models in a series. This reliance on the PCI-E interface necessitates an affordable and reliable means of testing and validation during the development of a prototype.

The primary focus of this project will be the validation of PCI-E interfaces. The validation of the physical layer of this interface presents the greatest challenge as it handles the physical transmission of data. The high transfer speeds of this interface increases its susceptibility to errors as this necessitates stricter signalling requirements. A simple tool that provides a remarkable amount of insight into the signal integrity of an interface is the eye diagram.

This project implements existing methods of eye diagram reconstruction to provide a simple and cost effective means of validating some of the interfaces present in Opengear's products. This was achieved through a synchronous reconstruction method implemented on an FPGA to test and validate the PCI-E interface. Additionally, a proof of concept implementation of an asynchronous eye reconstruction algorithm was developed in MATLAB. This was followed by a physical implementation that was able to reconstruct an eye from a generated pseudo-random bit sequence. Given appropriate measurement equipment, this asynchronous method will generate an eye diagram for any given interface. For future work the synchronous method could be expanded to test other high speed interfaces on the FPGA, or an ADC expansion card could be acquired for the FPGA and the asynchronous method could be incorporated with the developed system.

This final report offers the results of the development of this project. The report will provide an final overview of the motivations and scope of the project in Section 1. This is followed by a detailed investigation into the relevant theoretical background in Section 2. Details of the system design are presented in Section 3. Section 4 describes the testing methodologies and shows the data collected to validate the designs. Section 5 draws conclusions from the presented data and provides recommendations for future work.

Contents

List of Tables

List of Figures

Key Terms

AD2 Analog Discovery 2.

ADC Analog to Digital Converter.

AXI Advanced eXtensible Interface.

BER Bit Error Rate.

DRP Dynamic Reconfigurable Port.

DUT Device Under Test.

FIFO First-In-First-Out.

FPGA Field Programmable Gate Array.

I^2C Inter-Integrated Circuit.

IC Intergrated Circuit.

IEEE Institute of Electrical and Electronics Engineers.

JTAG Joint Test Action Group (Referring to IEEE 1149.7).

PCI Peripheral Component Interface.

PCI-E Peripheral Component Interface Express.

PCI-SIG Peripheral Component Interface Special Interest Group.

SATA Serial Advanced Technology Attachment.

USB Universal Serial Bus.

USB-IF Universal Serial Bus Implementers Forum.

1 Introduction

1.1 Overview

As demand for high data transfer rates increases [1], it becomes more important that the interfaces developed to support these rates are able to be tested and validated during implementation. Currently, integrity testing on such interfaces often requires expensive equipment and software. Although it is possible to seek an external provider to validate these interfaces [2]–[4], these providers usually only offer full interface compliance testing. This process is costly and often unnecessary for embedded systems where interfacing to general purpose hardware is not a requirement. This is especially true for the prototype phase of product development, where it is useful to be able to validate interface performance before continuing development with a specific design. As such, this project will seek to develop a system that will offer such insight quickly and affordably.

1.2 Context and Motivation

Opengear is a company that designs, manufactures, and sells console servers. In their larger product iterations the design of these console servers revolves around a main board common to all models in a series. These models are differentiated by various add-on boards that offer distinct functionality. These add-on boards are connected via a Peripheral Component Interface Express (PCI-E) interface. Currently, this interface undergoes functionality testing to establish whether it works but there is no further validation of the electrical performance of the interface. Due to the reliance on PCI-E it is important to have a cost-effective and reliable avenue for testing and validating this interface and, if possible, the other interfaces present.

1.2.1 Interface Compliance

There are multiple organisations whose role is to publish and update high speed interface standards, such as the Peripheral Component Interface Special Interest Group (PCI-SIG), Institute of Electrical and Electronics Engineers (IEEE), and Universal Serial Bus Implementers Forum (USB-IF). Through the publishing of these standards these organisations also provide the specifications by which these interfaces should be tested. For a device manufacturer to advertise that they are supporting a certified interface they must pass the suite of compliance tests [5], [6]. These tests typically cover the following:

- **The Physical Layer:** Logic voltage levels, signal timing requirements, and encoding schemes.
- **The Configuration Space:** Register addresses standard to an implementation that configure the functionality of the interface.
- **Transaction/Link Layers:** Layers of abstraction from the physical that allow interface control, reading, and writing.

However, it can be the case that these interfaces are not being used in total adherence to the standard, which is often the case in embedded hardware. In such a case it may not be necessary to test every aspect of the interface. For the purpose of this project the focus will be testing of the physical layer of an interface.

1.2.2 Current Solutions

Currently, a number of solutions exist to test and validate interfaces. Given the high speed nature of the signals that are being investigated the use of high speed and high bandwidth oscilloscopes, test fixtures, and specialised software are needed. Unfortunately, this equipment is often prohibitively expensive, typically costing tens of thousands of dollars [7]–[9] with a sample of costs collected from Keysight and the PCI-SIG shown in Table 1.1 below,

Item	Cost
Keysight Infiniium Oscilloscope	AUD 28,442
Keysight PCI-E Compliance Application	AUD 15,402
PCI-SIG Compliance Base Board	AUD 4,925
PCI-SIG Compliance Load Board	AUD 4,800
Total	**AUD 53,569**

Table 1.1: Sample Costs of Dedicated Test Equipment

Alternatively, a device may be sent to an external lab for verification, though this too is often costly and generally too time consuming to be a worthwhile option during development. A less expensive option is to verify the device at an industry event known as a “plugfest”. These events are conventions where equipment manufacturers test device interoperability, and validation/certification can be sought from the writers of industry standards for a cost. However, this option also lacks utility for developers wanting to assess prototypes that are still being worked on, and is not a particularly practical option as these events are infrequent, only taking place a few times a year.
Finally, a solution that does not require the use of an oscilloscope is for the developer to

practically test the product by testing whether the operating system detects the interface and an attached peripheral, and if the link is performing as expected. As mentioned previously, Opengear employs these functionality tests currently which are a practical solution. However, it does not take into account possible long term performance, nor does it present an indication of how well the interface is working, or any way to estimate product yields.

1.3 Scope

This project aims to implement a physical method of validation and characterisation for at least one of the interfaces used in Opengear's products. This will be achieved by collecting the information required for an eye diagram (See Section 2.3), so that at least one interface can be quickly and easily characterised and validated. This is to be accomplished with equipment already available within Opengear, including a Xilinx AC701 FPGA development board. After a review the stretch scope was extended to encompass a second goal. The proposed addition was later revealed to be not as useful to the project as initially thought, and was subsequently removed, leaving the stretch scope in it's original state. As such, the stretch goal of the project is to implement at minimum a proof of concept of an eye reconstruction algorithm, which would theoretically be able to construct an asynchronous eye diagram for other interfaces with equipment of the correct bandwidth. This method of testing would remain cheaper than the existing interface testing equipment mentioned in Section 1.2.2.

1.3.1 Objectives

The primary objective of this project is to deliver a system that produces an eye diagram of a PCI-E interface using synchronous methods. This primary objective was established as the minimum viable product that should be produced over the duration of the project. The secondary objective will be to produce a more general eye diagram reconstruction using asynchronous methods and could theoretically be used to test any interface given the appropriate measurement equipment. This secondary objective was established as an interesting proof of concept that, with the purchase of certain equipment, would be highly useful to Opengear. To address the needs of the scope and objectives a plan with task breakdown was developed. A detailed description of this task breakdown and the planning methodology is shown in Appendix B.

1.3.2 Requirements

This project aims to develop multiple features for Opengear. The following requirements are necessary for the successful development of the project product, as they primarily focus on utility for Opengear.

1. **Synchronous Eye Measurement System**
   A system that utilises a Field Programmable Gate Array (FPGA) board connected to a PC using simple software to reconstruct the eye diagram of the PCI-E interface that the FPGA is connected to.
2. **Stretch: New FPGA design with PCI-E loopback (Removed)**
   A new FPGA design that implements a PCI-E loopback First-In-First-Out (FIFO) buffer that allows the eye diagram to be measured while the interface is under different loads. This stretch goal has been removed from scope as it was decided to no longer be valuable.
3. **Stretch: Asynchronous Eye Reconstruction**
   A proof of concept of an algorithm that can asynchronously reconstruct the eye diagram of an interface through undersampling, allowing a physical implementation to generate an eye of any measured interface.

1.3.3 Deliverables

Concurrent to the requirements of the finished project listed above there are a number of academic deliverables that are needed for the successful completion of the course that this project pertains to. These deliverables have been listed below.

1. **Project Proposal (Delivered)**
   The project proposal will detail the motivations for this project and draft a plan for the project's execution.
2. **Interim Report (Delivered)**
   This report will review the progress of the project and other relevant information including but not limited to the goals, relevance, literature review, and risks.
3. **Oral Presentation (Delivered)**
   A presentation of the final work and possible final stretch goals to the academic staff, other students from the course, and placement hosts.
4. **Final Report (This Document)**
   A final report containing all of the results, comments on the outcomes and, if applicable, recommendations for future work.
5. **Online Reflections (Delivered)**
   Reflections on the progress of the project and personal development relating to the Engineers Australia Stage 1 Competencies.

2 Literature Review

This literature review will detail the background information relating to the current scope of the project. Roughly following the steps that led to this scope, it will also provide more depth relating to the motivations of the project. Summarily, the following sections will detail:

- The specifics of certain interfaces.
- Factors that affect signal integrity.
- A detailed look at eye diagrams.
- Methods of eye diagram reconstruction, both synchronous and asynchronous.
- The relevant properties of ADCs.
- An overview of FPGA's and their associated technologies.

2.1 High speed interfaces

Interface	Number of Pins	Connection Style	Data Rate
I^2C [10]	2	Serial	425 kB/s
PCI-E 4.0 [11]	26+(8 per lane)	Serial	1.969 GB/s (per lane)
PCI [12]	124 (32 bit)/188 (64 bit)	Parallel	133 MB/s
USB 3.0 [13]	9 (Type A)	Serial	625 MB/s
SATA 3.0 [14]	22 (standard)	Serial	600 MB/s
Ethernet (10GE) [15]	8 (cat6)	Serial	1.25 GB/s
RS485 [16]	2 (per RX/TX signal)	Serial	125 kB/s

Table 2.1: Comparison of Common High Speed Interfaces

There are multiple standards that have been introduced for internal and external peripheral connections (See Table 2.1 for comparison of current revisions of interfaces), but the most prevailing are two serial standards: PCI-E and USB. These interfaces will be investigated in greater depth as they will be the focus of this project. The PCI-E interface

was introduced in 2003 to replace the older parallel PCI standard and has become widely used for high speed internal peripheral connections to CPUs [17]. This is part of a larger trend in computing away from parallel interfaces and toward serial ones. USB follows this trend, though released much earlier in 1996, replacing multiple parallel interfaces [18]. This is due to the nature of the timing requirements needed by parallel interfaces, imposing complex design challenges on increasing their data rates.

2.1.1 PCI-Express

The PCI-E interface provides a serial communications link between its endpoints, and switches are often used to allow one link to support multiple endpoints. These links are organised into lanes where each lane supports one of these links and a single endpoint can communicate on a number of these lanes simultaneously [17]. The interface is composed of three layers of abstraction, the transaction layer, the data link layer, and the physical layer [19].

The physical layer is the electrical connection between two endpoints and handles the physical transmission and reception of packets. Each lane between the endpoints consists of two differential signalling pairs which form a full duplex serial link [19]. Additionally, the physical layer incorporates the serialisation and deserialisation logic of each lane. This layer also supports the protocol for link training which is the initial establishment and negotiation of certain link parameters. PCI-E supports transfer speeds of 16GT/s per lane, and up to 32 lanes can be utilised between two endpoints [11].

The data link layer manages the reliable delivery and reassembly of packets across a lane. It does this through error checking and in the case where multiple lanes are implemented, removing timing skew between lanes [20]. Error correction is maintained by a replay buffer which requests a packet resend if an error is detected [19]. These requests are handled by specific data link layer packets which act as control messages between two endpoints. If the errors cannot be corrected then the link is reset and retrained [20].

The transaction layer provides the interface to and from the host operating system to the link itself. This layer accepts and buffers packets received from the link layer so that it can provide meaningful messages to and from the interface [17]. It also constructs and sends packets to the link layer based on requests made in software from the host. Finally, it has mechanisms for flow control and managing the ordering of packets sent on the link [19].

2.1.2 Universal Serial Bus

USB provides a standardised means of connecting the vast majority of peripheral devices. For the scope of this project, USB provides a means for developing a Host/Device relationship between two endpoints using a serial communication link or links, depending on the revision of the standard being used [21], [22]. The specification also has certain power delivery components to it, however these are not relevant and will be considered outside of the scope of this project. The USB interface is also comprised of three layers of abstraction, the physical layer, the link layer, and the protocol layer [18].

The physical layer of USB provides a similar function to that of PCI-E. The physical link is made up of a single differential pair forming a half-duplex link for USB 1.0-2.0, with an additional two differential pairs that serve as a full-duplex communication channel for USB 3.0 and up [13]. The original link allows for bit rates up to 480Mbps and provides the mechanism for devices to communicate when they are in a low power state, allowing one side of the link to wake the other if needed [21]. The additional communication link added in revision 3.0 effectively allows the 2.0 link to manage the control aspects of the overall connection while utilising the high bit rate of the 3.0 link (5Gbps) to handle the actual data transfer [13]. Much like PCI-E this layer also handles the serialisation and deserialisation of the data before it is transmitted, and for USB the entirety of the physical layer functionality is often contained in a single chip called a PHY.

The primary functions of the USB link layer are managing and controlling the link. Unlike PCI-E this layer in USB handles the initial link training [23]. This control is continued throughout operation as it manages power states, error handling, and fault recovery.

The protocol layer for USB ensures the flow of packets between two endpoints. In conjunction with the link layer it handles packet transmission structure and synchronisation between two endpoints [13]. Similar to the transaction layer of PCI-E, it also specifies a set of packets that can be sent across the link allowing for layers of software to interact with the link [23]. This layer also handles the calculation of error correction codes for each packet along with packet format.

2.2 Signal Integrity

The validation of high speed interfaces becomes increasingly difficult and complex as higher bit rates are achieved. These gains come with an increase in the interfaces susceptibility to errors, as timing requirements and logic voltage levels have to be more strictly defined. As seen in Section 2.1, these interfaces typically have layers of abstraction and some errors can be seen and corrected at higher layers closer to software. However, errors in timing and noise are often due to how a design has incorporated the interface and

these require different means to be measured and corrected. The main contributors to the degradation of signal integrity on a communications link are jitter and noise.

Jitter is a very complex issue and there are a number of different definitions, even amongst IEEE standards [24]–[26]. Though the specific definition of jitter in the context of this project is not crucial, the definition used will be taken from the ITU [27] as "The short-term variations between the optimum sampling instants of a digital signal and sampling clock derived from it." As clock frequency drift is later defined as the cumulative effect of these variations, it is assumed this definition of jitter refers to non-cumulative variations. At high speeds, if jitter is substantial then transmitted data could appear out of place, or order. This can corrupt the packet being sent, therefore, it is critical to be able to mitigate this jitter. The resultant transmission errors, if frequent enough, can cause the link to reset and retrain, interrupting use. An example of jitter is illustrated below in Figure 2.1.

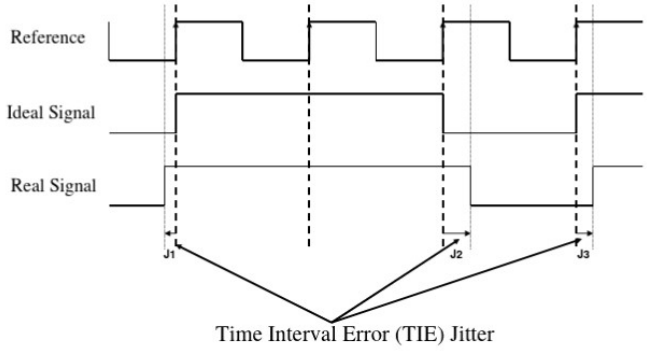

Figure 2.1: Illustrated definition of jitter [28]

Another important factor when considering signal integrity is noise. Noise can usually be defined as unwanted variations or disturbances in a signal, and although jitter can be regarded as frequency noise, in this case it is amplitude noise that will be considered. There are a number of places that noise can originate from in a communications link, such as the transmitter, the cable or interface, and the receiver. This noise can prove highly detrimental to signal integrity as when unmitigated it can cause bit errors as interference causes a 1 to be read as a 0 or vice versa. The Bit Error Rate (BER) of a link is an often used metric when determining its quality.

An interesting technique to measure and observe both noise and jitter in a signal is the eye diagram. The eye diagram is a relatively simple tool that provides a remarkable amount of insight into the integrity of a communications link. In some cases BER can be measured simultaneously to collecting data for an eye diagram or, alternatively, the BER can be extrapolated from the same data.

2.3 Eye diagrams

As mentioned in the previous section, eye diagrams are a useful tool for measuring various aspects of a communications link. Not only does an eye diagram present a useful visual aid, it also provides extensive information about a link. This information includes: estimated BER and SNR, rise time, fall time, and jitter. This can all be used to characterise the integrity and performance of a link, all useful during the prototyping phase of product development.

An eye diagram is constructed by measuring a series of bit transitions on a link. Then, typically by knowing the average interval over which a single bit is transmitted, overlaying all of these bit transitions on top of each other. Ideally, this overlay would just show a single shape, the eye of the eye diagram, seen in Figure 2.2. However, in reality jitter and noise have an effect on a signal and it results in numerous visible traces of bit transitions, seen in Figure 2.3.

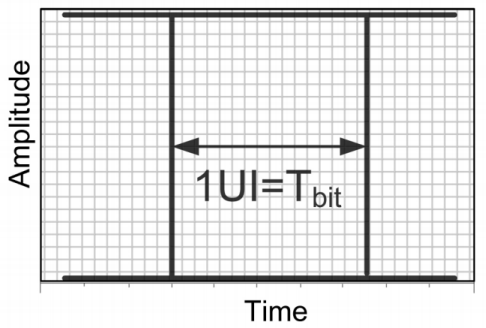

Figure 2.2: Ideal Eye Diagram [29]

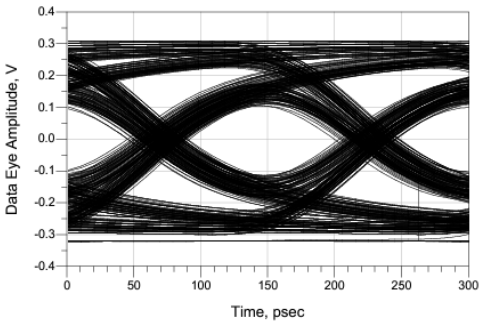

Figure 2.3: Typical Eye Diagram [30]

Often in compliance tests a specific bit pattern will be transmitted while the data is measured for an eye diagram. This is to ensure that all of the necessary bit transitions are recorded and the most accurate version of the eye is attained. For less formal eye diagrams used to estimate link quality, the data can be measured over a longer period of time during normal link operation. This large volume of data increases the probability that the result will produce the same information without the need for a specific test pattern to be transmitted over the link.

2.4 Eye Reconstruction

Eye diagram reconstruction can be approached in a number of ways which can be broadly categorised into synchronous and asynchronous methods. Synchronous methods involve the extraction of clock data from a signal and are typically implemented at a circuit board and integrated circuit level due to their complexity. Asynchronous methods are less developed and rely heavily on digital signal processing techniques. They primarily

focus on estimating an alias of the bit frequency which is then used to reconstruct the eye. This is beneficial as it is not then necessary to have knowledge of the frequency of the signal being analysed. In turn this moves the complexity out of hardware and into software.

2.4.1 Synchronous Methods

Synchronous methods of eye reconstruction require knowledge of the clock signal of the signal being investigated. When a clock signal is not transmitted separately over an interface, a clock and data recovery module is used to extract this clock signal, which can then be used to sample the data that was transmitted [31]. The complexity and accuracy of design in these modules becomes increasingly relevant as the speed of the interface increases. If the transmitted signal is directly sampled the recovered clock signal can be used to overlay these samples within an average interval to build an eye diagram. Another method is to compare the signal to a sweeping reference voltage over each clock cycle [32]. This establishes a distribution of signal levels at different points in the clock cycle which can be used to derive an eye diagram, given enough samples. The primary advantage of these approaches is that they can often be implemented in hardware [31], are relatively simple in their design, and may not need much post processing. A disadvantage of this is that these systems often need to be designed into transceiver circuits, which is not always appropriate, or alternatively they can be tested using an Analog to Digital Converter (ADC) with a high bandwidth and sample rate to directly sample the signal, though this is an expensive solution.

2.4.2 Asynchronous Methods

Asynchronous methods of eye diagram reconstruction do not require a knowledge of the clock or bit rate of the incoming signal. Most asynchronous methods undersample a signal and then do significant processing on the data to reconstruct an eye diagram. The main focus in these methods is recovering or estimating an alias of the bit frequency in order to reconstruct the eye diagram over a normalised unit interval. The methods described in [33] focus heavily on recovering this alias, and have been validated on highly undersampled signals.

Recovering the frequency spectrum of the undersampled signal shows only noise, so no useful information can typically be recovered from this raw data. The research found that by applying a non-linear transform that emphasises data captured in the transitions between bits some information could be recovered. When this transformed data is used to generate a periodogram a good estimate of the aliased bit frequency is able to be resolved. This aliased bit frequency can then be used in a series of correctly windowed

pseudo-Fourier transforms to apply phase shifts to each of the data points in the original sample. This series of steps after finding the alias can be simplified down to passing the non-linearly transformed data through a specially designed FIR filter.

Another method for estimating the aliased bit frequency is proposed in [34] . This solution effectively estimates the aliased frequency by passing the sampling clock and the incoming signal, after some arithmetic division, into a designed estimator component. This is able to provide a relatively accurate estimate of an aliased bit frequency. This is then used to reconstruct the eye using the same method as [33]. The advantages of these methods are that they require no knowledge of the interface and could theoretically provide an eye diagram for any interface given sufficient sub-sampling. However, the signal must still be sampled with a high bandwidth ADC, though is a better solution than synchronous methods as they also require a high sample rate, which asynchronous methods do not.

2.5 Analog to Digital Converters (ADCs)

In both of the standard methods of eye reconstruction there is a need to directly sample the incoming signal. This can prove to be difficult for the high speed context of this project as the signal to be sampled can be running at multi-gigabit speeds. There are a number of different types of ADC architectures but most incorporate a stage of sample and hold. For an ADC to correctly sample a signal running at these speeds, it would need to have certain appropriate parameters. These primarily being, aperture and acquisition time, sampling rate, and bandwidth [35]. These parameters determine the maximum frequency of a signal they can sample and how well it can be sampled. The details of each of these parameters will be expanded upon below.

2.5.1 Aperture and Acquisition Time

In sample and hold architecture there must be a switch that is driven by the sampling clock that allows the input signal into the hold capacitor [36]. The aperture time is defined as the amount of time between the system sending the hold command and this switch disconnecting the input to the capacitor. Closely related to this is the acquisition time which is the amount of time it takes for the hold capacitor to charge. Both of these values dictate the speed of the signal that can be sampled. If the signal is changing too quickly the hold capacitor may not capture an instant in time but instead receive a largely changing voltage [37]. This can lead to large inaccuracies in the output of the ADC.

2.5.2 Sampling Rate

The specifications for an ADC will always include the sampling rate. This sample rate typically specifies the maximum number of output samples in a unit of time. Often there will also be a specified minimum sample rate [35]. This is due to the hold capacitor, as it is selected to handle the fastest incoming signal. Low acquisition time comes at the cost of low hold time which means it will not be able to sample a signal correctly below the specified minimum [38]. This is not as much of a problem in the use case discussed of extreme undersampling. However, it is still to be considered as it is often referenced early in the datasheet and can give an insight into some of the other parameters such as those mentioned above.

2.5.3 Bandwidth

As a result of the properties above there are limits on the frequencies that can be correctly sampled. The bandwidth specifies the the frequency where the voltage gain of the input architecture has experienced a 3dB drop [37]. Alternatively, the bandwidth can also be represented as a function of the sample and hold slew rate. The slew rate is defined as the maximum rate of change in the output voltage when the sample and hold architecture is in sample mode [36]. This depends heavily on the hold capacitor which links the choice back to acquisition and aperture time. It then becomes important to verify all of the properties of the ADC when selecting one for an application.

2.6 Field Programmable Gate Array (FPGA)

An FPGA is a re-configurable Intergrated Circuit (IC) that can be programmed to have a particular functionality. A typical FPGA includes a number of configurable logic blocks, input-output buffers, digital signals processing slices, and block RAM [39]. This is combined with a general routing structure to connect these individual components [40]. The programming of an FPGA is done via a hardware descriptor language, commonly VHDL or Verilog. One of the primary benefits of using an FPGA produced by a prominent manufacturer is that they often provide a number of common and complex functions in the form of Intellectual Property (IP) cores or blocks. These can range from a simple FIFO buffer to a fully implemented microcontroller. Furthermore, development boards released by these manufacturers typically come with a number of extra components that can be interacted with. These can include LCD screens, gigabit transceivers, external RAM, and daughter card attachment ports [41]. The Xilinx 7-Series Artix FPGA development board (AC701) implements the 7-Series GTP transceivers, some of which are used to implement a PCI-E interface. These transceivers, which can be accessed through the

Dynamic Reconfigurable Port (DRP) interface, can be reprogrammed during operation and are the primary reason this board was chosen for the project. The details of these and other components relevant to the project will be explored further in the following subsections.

2.6.1 GTP Transceivers

The GTP transceiver is a gigabit transceiver designed by Xilinx to be used in conjunction with the 7-Series FPGAs. This specific transceiver was designed to provide the highest performance per watt at one of the lowest prices in the 7-Series, and supports transmission rates between 500Mbps and 6.6Gbps [32]. Given these supported data rates these transceivers are able to support the physical layer implementation of a number of standard high speed interfaces such as PCI-E 1.1/2.0, DisplayPort, SATA, Gigabit Ethernet and many others [42]. Each transceiver channel is comprised of two differential pairs, one for transmitting and one for receiving. Four of these channels are then organised into a Quad which shares two differential reference clock pairs, analog supply pins, and two ring oscillators to drive the IO for the channels [32]. This description is better understood in Figure 2.4 below,

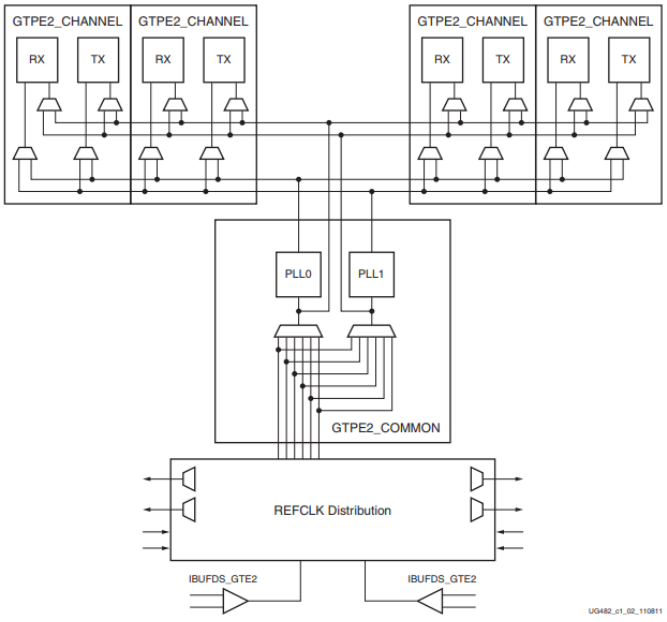

Figure 2.4: GTP Quad Layout [32]

One of the impressive features of the 7-Series transceivers is their ability to perform limited signal analysis whilst they are transmitting normal data. This can be done by adding another signal path with programmable phase and voltage offsets that then leads to an secondary sampler. By doing this the difference between the offset signal and the original signal can be compared for all the different combinations of offsets. Comparing

the number of differences between the offset signal and the original, and the total number of bits received can give the BER. Similarly, for a given pattern of bits being received the original and offset signals can be compared for differences to produce a waveform eye diagram view to give a general idea of the integrity of the signal being received. The Dynamic Reconfigurable Port (DRP) interface makes this possible.

2.6.2 DRP Interface

The DRP allows the parameters of any given channel of the GTP to be changed dynamically. The DRP interface is a synchronous memory mapped interface with an address bus and separate data busses for reading and writing. These are used in tandem with a number of control signals that are used to indicate data availability, operation completion, along with read and write requests [32]. Although this is a simple enough interface for accessing the configuration and storage registers of the transceivers, it is not supported by the Xilinx microcontroller IP. This means that a bridging interface must be used to provide access to these registers [43]. A common method of doing this is through an Advanced eXtensible Interface (AXI) to DRP bridge. This provides access to the full range of addresses on the transceiver to an AXI master, which appears in the microcontroller IP core.

2.6.3 AXI Interface

The Advanced Microcontroller Bus Architecture is a set of specifications for the interconnection and management of and between controllers and peripherals that appear in silicon and FPGA design. In the more recent revisions the primary architecture specification is that of AXI4 which details three standards for interconnections: AXI4, AXI4-Lite, and AXI-Stream [44]. The AXI4 protocol is suited to high frequency, low latency systems providing a high performance memory mapped interface. The AXI4-Lite interface is a lightweight version of the of the AXI4 interface for simpler low throughput interfaces. Finally the AXI4-Stream interface removes some of the restrictions of AXI4, primarily it eliminates the limits on data burst size, enabling for high-speed data streaming.
The AXI4 and AXI4-Lite interfaces consist of five different signalling channels, including: read and write address channels, read and write data channels, and a write response channel [45]. All of these channels also contain *ready* and *valid* signals for handshaking, and some might come with extra control signals. For a basic *ready*/*valid* handshake one side of the interface will assert *valid* high and wait until the other asserts *ready* high before transmitting. To transfer data on the interface these channels use these *ready* and *valid* signals to perform a handshake in order to transfer the address that the data will be sent to, and then begin the transfer of data [44]. Once this initial handshake is complete,

AXI4 can use some of its other control signals to complete up to 256 data transfers to the same address [45]. The stripped down AXI4-Lite interface on the other hand, can only complete one data transfer at a time.
The AXI4-Stream interface on the other hand, is more useful when mapped address spaces are not needed. In general, this protocol performs a single *ready/valid* handshake, transferring no address information, and then begins the transfer of data [46]. However, if it is assumed that the receiver is always ready to receive data the *ready* line can be omitted. There is no limit to the number of data transfers that can be completed following the initial handshake for AXI4-Stream.
Xilinx supplies a number of IP cores that provide interconnects for each of these protocols to allow a number of master and slave devices to be connected [45]. There are also IP cores available that allow the interconnection of different AXI interfaces, and the interconnection of AXI and other memory mapped interfaces such as DRP. This project implements AXI4, and one of the stretch goals would seek to use an AXI4-Stream interface.

2.6.4 MicroBlaze IP Core

The MicroBlaze IP Core implements a fully functional microprocessor that has been optimised for use on Xilinx FPGAs [47]. This is a "soft" microprocessor, meaning that is is fully implemented using the programmable logic of an FPGA and is therefore highly customisable. The IP also includes an IO module, a standard set of peripherals and local memory. The memory implemented by this module can be used to store programs to be run on the MicroBlaze and store collected data. The main versatility of the MicroBlaze is that its primary form of IO is an AXI interconnect [47]. This allows it to act as both an upstream master to peripherals such as a DRP bridge, or as a downstream slave, receiving data from components like a JTAG interface [43], [48]. All of these components of the MicroBlaze can be customised to suit the needs of the application.

3 Project Description

This section will provide a detailed description of the PCI-E scanning system and the asynchronous eye reconstruction algorithm implementation. The main focus will be to provide details of the various components that form the whole of both of these systems. The primary objective will be broken into its main components of hardware and software. These will then be broken up further to cover all of the relevant sub-components relating to the function of the system. This will be followed by the stretch goal which will primarily focus on the description of the algorithm and its implementation.

3.1 PCI-E Scan System Design Description

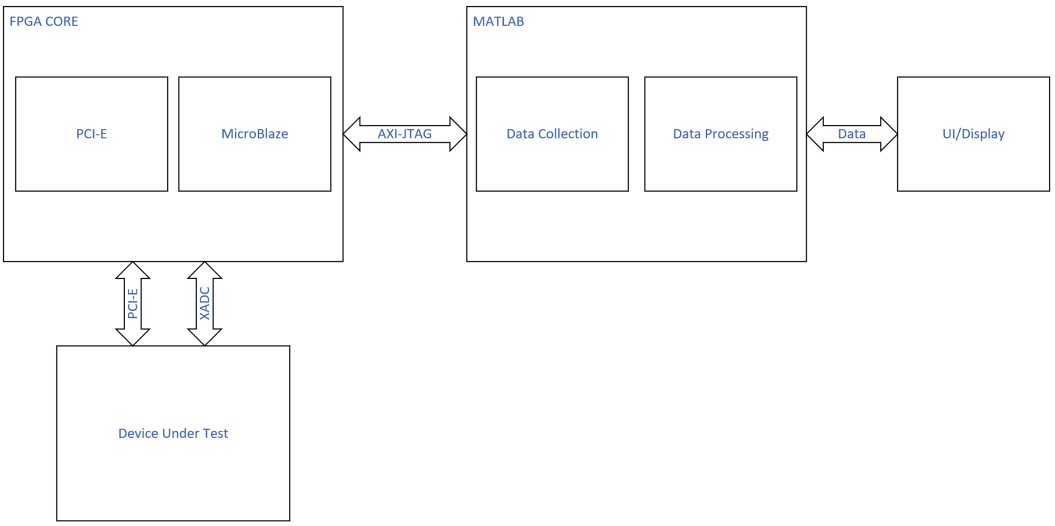

Figure 3.1: Block Diagram of the Designed System

Figure 3.1 shows a high level overview of the final design of PCI-E scan system. This system implements the components outlined in both the initial proposal and the interim report [49], [50]. The FPGA Core component acts as the main scanning mechanism which configures the available GTP transceivers to perform either a BER or waveform eye scan on the PCI-E interface. Xilinx application note 1198 [51] was used as the basis of the FPGA architecture though it had to be heavily modified to perform the waveform scan. The data from these scans is sent to the MATLAB component of the system which processes and displays this as an eye diagram. This processed data is then analysed to produce some limited quantitative measurements of eye compliance. This is all contained

within the user interface which provides the functionality to store and load previous scans and to generally interact with the scanning process.

3.1.1 FPGA Core

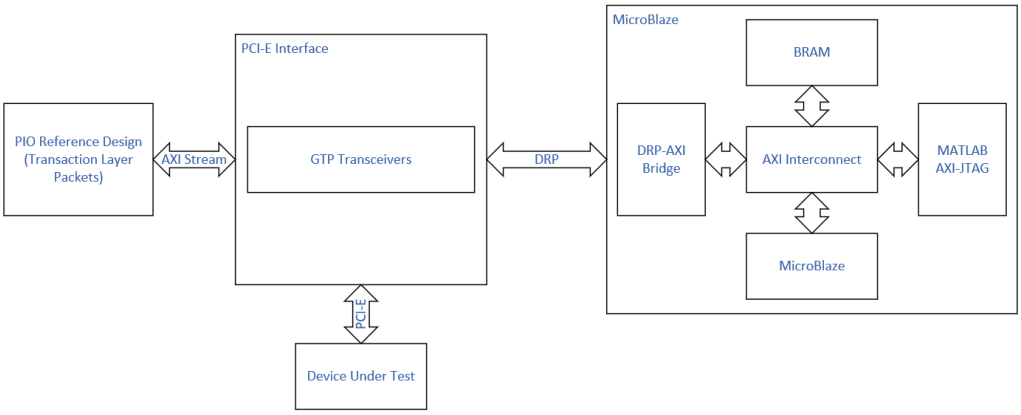

Figure 3.2: Block Diagram of the FPGA Core

Figure 3.2 shows the detailed block diagram for the FPGA core. This houses a MicroBlaze which handles the majority of the communications between the FPGA and the PC running MATLAB. This communication is entirely done through the AXI. The MicroBlaze also manages the collection and storage of data attained from the DRPs. The FPGA also houses the IP Blocks for managing the PCI-E interface. This creates the link between the Device Under Test (DUT) and the FPGA. Figure 3.3 shows the resource utilisation of the FPGA. It can be seen that although the PCI-E uses half of the transceivers the rest of the implementation is quite minimal.

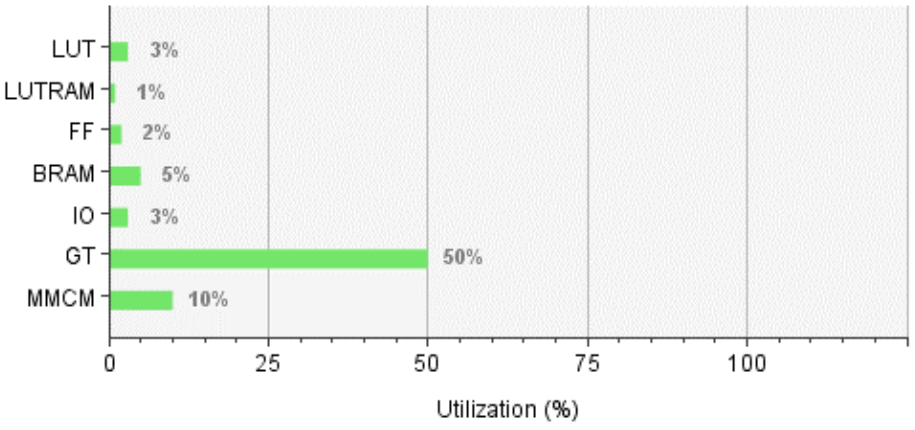

Figure 3.3: FPGA Utilisation

MicroBlaze

The MicroBlaze used in this implementation is a significant alteration of the reference design that appears in Xilinx application note 1198 [51] which is an implementation that focuses entirely on the collection of data for a BER eyescan. This implementation had to be altered to accommodate the new mode of scanning to get an approximate waveform eye diagram. In the BER implementation the transceiver would apply and increment phase and voltage offsets to the incoming signal before it is sampled and received. This offset signal is then compared to the original, then a count of samples received and number of times the offset signal and original signal differed are recorded. The new scanning method applies the same offsets, however it looks for a predefined pattern in the original signal. When there is a match both the data sampled from the offset signal and the original signal are recorded. Each lane of the PCI-E has a specific section of memory that stores both its configuration and the data collected from the transceiver, when data fills this section for different combinations of voltage and phase offsets the data is transmitted as a block back through the MATLAB-JTAG.

PCI-E Interface

The design implements a PCI-E interface using a combination of a Xilinx IP and reference design. An AXI stream connected IP is used for the implementation of the PCI-E interface itself. This provides the Microblaze with its clocking signal as it also provides the connection for the AXI-DRP bridge which must be the same. This IP also performs the initial bring-up and link training for the interface. The implemented reference design is a stripped down version of the memory end point tester seen in Xilinx application note 1022 [52]. This provides a constant stream of valid PCI-E transaction layer packets requesting basic memory reads from the other side of the interface. This provides a constant stream of responses back to the FPGA that can then be received used for the analysis that produces the eye diagram.

AXI Interconnect

The AXI plays an integral role in the overall design. The AXI interconnect allows the various components shown in Figure 3.1 to be connected to the single AXI master port on the MicroBlaze. This is obviously important to the overall ability for the design to function. More importantly the interconnect allows more than one master to be connected. This means that the MATLAB-JTAG connection can also act as a master allowing it to pull the data directly from the BRAM when it is ready rather than having the MicroBlaze manage bulk transmission of data. Instead the MATLAB-JTAG is also connected as a slave so that the MicroBlaze can send a control signal when the BRAM is ready to be accessed.

3.1.2 MATLAB

The MATLAB software is presented as a standalone executable program that can be installed on any windows computer. It succeeds in implementing the features that were laid out in the initial proposal and interim reports. These features primarily focus on the collection and storage of data from the FPGA, the processing of this data into an eye diagram display, and the measurement of certain features of the displayed eye diagram. This program also provides a user interface for the customisation of certain scan parameters. The following sections will provide a detailed description of the mentioned features. The MATLAB code is cleanly divided into four sections: data collection, BER data processing, waveform data processing, and parameter measurement.

Data Collection

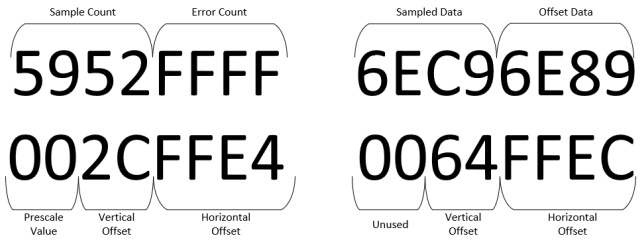

Figure 3.4: **Left:** Example BER Data, **Right:** Example Waveform Data

The original reference design for the collection of BER data used code written in the tcl scripting language specifically to interface with the Xilinx Vivado Design Suite. This was used as a reference to perform a similar task in MATLAB through a different interface. To do this, a JTAG connection is established with the FPGA and the relevant scan configuration information is sent. A file is created for each lane of the PCI-E being scanned, and the raw data is stored there as it is received in blocks from the FPGA. The data received contains the offset values for that section of the scan and then either the sample and error counts, along with a prescale value, or the raw data from the interface all represented as four bytes of hexadecimal data per data point (A comparison can be seen in Figure 3.4). This data is then split and converted into different arrays indexed to correspond to the same data points.

BER Data Processing

For BER data processing the sample and error counts are extracted from the data, along with the prescale value, which acts as a multiplier on the sample count as it would be

too large to store normally (it also determines the scale of the BER diagram). With this data the diagram is constructed by using the following equation.

$$BER = log\left(\frac{errorCount}{sampleCount \times dataWidth \times 2^{prescale+1}}\right)$$

Where dataWidth is the number of bits the transceiver compares at a time. An example of the diagram can be seen in Figure 3.5. Whilst the algorithm was closely derived from a Xilinx reference design [51], the display layer was originally handled internally within Vivado. Using the reference as a guide this was implemented in MATLAB, with the display layer being added to the process.

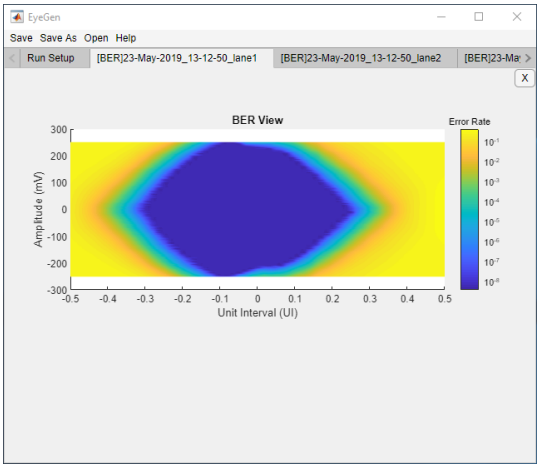

Figure 3.5: Example BER Scan Result

The standard for PCI-E includes various measures of a BER eye diagram. However, these measures are for a measured error rate of at least 10^{-12}. Using this measurement system that would take an unrealistic amount of time. The BER in this case is measure to 10^{-8} so that it can be viewed in a qualitative sense rather than the more quantitative measures presented by the waveform measures.

Waveform Data Processing

For the waveform view, a significant amount of work had to be done to process the data so that it would display a desirable result. This is because this mode of use is not officially supported by Xilinx, and minimal description of its implementation is provided. The method developed likely shows the regions of most instability in the signal which by extension shows a rudimentary eye diagram. For the transceiver to capture both the original signal sample and the sample taken after the offsets have been applied, a qualifier

pattern must be used. As the range of phase offsets is -1:1 unit intervals from the central sampling point, the qualifier pattern chosen was one that modelled all of the possible bit transitions over this range, while keeping the number of ones and zeroes equal (two example cases are '0xE8C' and '0x746'). The two samples are recorded and the following equation is applied,

$$waveform = \left| \frac{sampleCountOnes}{offsetCountOnes} - \frac{sampleCountZeroes}{offsetCountZeroes} \right|$$

Although this may seem unusual it can be seen that this would show a value of zero or close to zero within the opening of the eye, as this is the area of optimal sampling so the offsets should not affect the received signal. However, in the areas where the signals are more likely to be different, this would show non zero values as the sampler is not optimised for these regions. An example of the output diagram can be seen below in Figure 3.6. In this diagram the colour scale represents the likelihood of the difference where yellow shows the highest likelihood for difference and the blue in the centre shows that the samples were the same. Given the equation above is not exactly probabilistic the values are not shown and the areas transitioning between the solid colours represents more uncertainty in the scan.

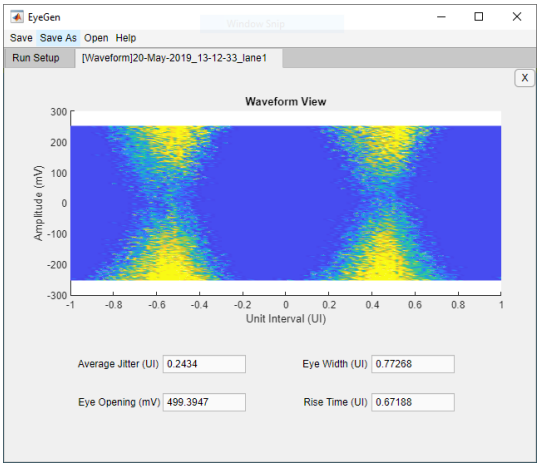

Figure 3.6: Example Waveform Scan Result

Parameter Measurement

The final piece of processing that is completed is the measurement of the properties of the eye diagram. The method chosen to perform this task was to find the average distance across areas of the graph to get a fair approximation. This is done by selecting a start

point on the graph and a direction. From this start point the program makes steps in the specified direction until it starts receiving errors that are ten percent of the maximum error rate. This was chosen because the area in the centre of the logic transition shows uncertainty, but due to the nature of the eye diagram this is an important area for measuring the width and jitter of the signal. Therefore the uncertainty is kept as part of the measurement because it offers valuable information about the nature of the signal. The distance measure is then repeated 10mV above and below the original start point. Any outliers are discarded and then the average is calculated. This is done to avoid any anomalies in the graph that would allow the distance measure to continue through to the other side of the graph. As an example, to measure the width of the origin would be chosen as the start point (the centre of the eye). This process would measure the distance in the positive phase direction and then again in the negative phase direction. These distances are then added together to give the total width of the eye. This is then repeated for the other measures which can be seen in Figure 3.6. In some cases there have been vertical bar artefacts on the left and right edges of the scan, the origin of these is unknown and their presence is seemingly random. These are accommodated for by shifting the measurement boundaries to exclude them when present. There is mouse over text on each of the text boxes displaying the parameter measures. This mouse over test shows the values required by the PCI-E 1.0 and 2.0 standards for compliance, these values were retrieved from a Tektronix document [53] that summarises the compliance requirements outlined in the PCI-SIG documents. These values are shown below in Table 3.1.

Parameter	Compliance Value
Average Jitter	Value $<$ 0.3 UI
Eye Opening	Value $>$ 175 mV
Eye Width	Value $>$ 0.4 UI
Rise Time	Value $>$ 0.15 UI

Table 3.1: Parameter Compliance Values

UI/Display

This interface was developed using MATLABs “appdesigner” toolbox. This allowed for rapid production of a GUI whilst being able to easily integrate the code-base that had been developed in the previous stages of the project. A number of minor interface updates were made on request of the engineers who would likely use the program. These features were not originally intended, but include a help page and a number of hot-keys for selecting different configuration options. All of the necessary configuration options that

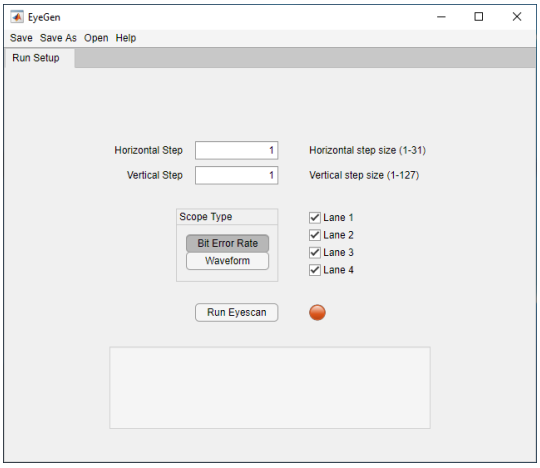

Figure 3.7: Configuration Tab of the GUI

are needed for the scan have been left configurable to the user. The options to save and load scans have been implemented, such that the files must conform to a simple standard that includes the configuration information followed by a raw dump of the collected data. Otherwise, scan data is stored temporarily and if it is not saved it is deleted upon exiting the application. The display tabs are dynamically generated as needed for the scans that have been completed or have been requested to be loaded in from storage. The interface for the different tabs can be seen in Figures 3.5 and 3.6, and the configuration tab can be seen above in Figure 3.7.

Device Under Test

This is the device to have its interface tested. This requires the FPGA to be connected to the PCI-E interface of the device to be scanned. This connection also has the following requirements,

1. At least a x4 PCI-E connector or,
2. An extension cable or adaptor that allows the x4 connection to the FPGA*
3. A fully implemented PCI-E interface that can attempt to establish a link

*Extensions or adaptors may impact the performance of the interface negatively.

3.2 Asynchronous Eye Reconstruction

This section will follow the method of implementation of the algorithm described in [33]. This algorithm was implemented in both MATLAB and python, given the similarities between the two when using the “numpy” module in Python [54], [55] this description

will cover the initial proof of concept done in MATLAB. References to code in this section are in reference to Appendix A.1. The steps of the process will be elaborated along side a plot of the relevant data obtained at each step. The data used for this description was generated in MATLAB with some random jitter and simulated to pass through a noisy channel to give as realistic a signal as possible.

1. Initially the raw data has a non-linear transform applied to it, as shown in line 3.

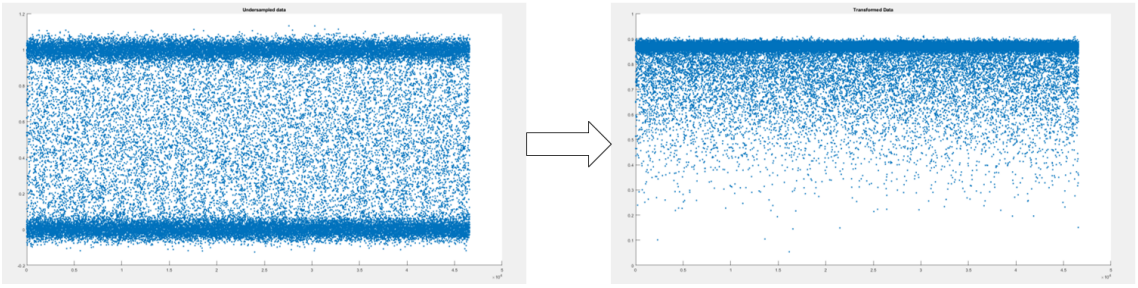

Figure 3.8: Effect of Data Transform

2. This transformed data is then cut into blocks of length 512. This was chosen to ensure speed for the repeated Fourier transforms that need to be calculated in the later stages of the process. To account for arbitrary data some data may be truncated from the end of the set to give an integer number of complete blocks. This data is then reshaped into an array 512 long in one dimension and the calculated multiple in the other, and then transposed so that the periodogram can be calculated correctly. This process is shown in lines 6-14.

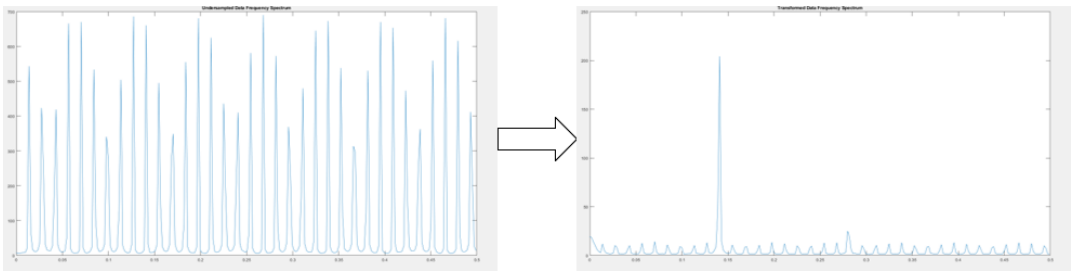

Figure 3.9: Effect of Data Transform

3. Due to symmetry only the lower half of the spectrum is investigated to find the peak frequency. This is done by using the built in peak finding functions of MATLAB, shown in lines 17-19. The value *wp* is the aliased bit frequency.

4. This is then used to calculate the correct phase shift to reconstruct the eye. This is achieved by taking a window of samples around the sample you want to correct. A Blackman windowing function was chosen because it reduces the effect of "less local" data points on the current phase calculation, which ensures that the eye diagram is centred correctly. A different window function with a similar effect could be used but the Blackman was found to perform correctly. This window of data is then Fourier transformed holding the frequency constant at the aliased bit frequency, shown in lines 22-37.

5. These phase shifts can then be plotted with their corresponding points of undersampled data to give the reconstructed eye diagram.

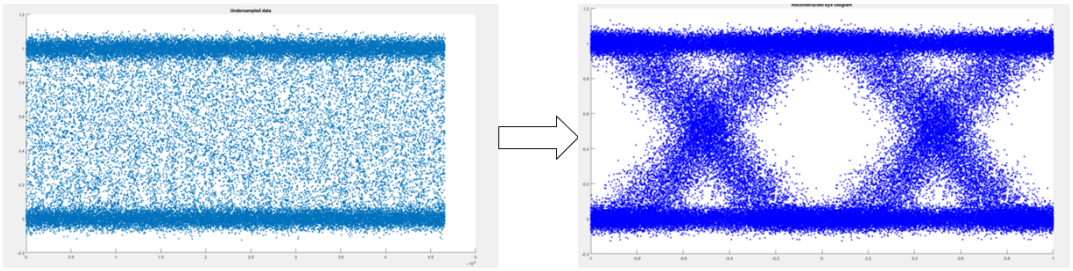

Figure 3.10: Eye Reconstruction

4 Test Methodology and Results

4.1 Test Methodologies

This chapter will detail the methodologies used to test the various methods of eye reconstruction. These were chosen to validate that each of the systems are able to correctly classify the integrity of the interfaces being tested. These tests were designed to provide this measure of validity using available equipment.

4.1.1 PCI-E Interface Test Methodology

A series of tests were run to demonstrate that the designed system performed to an acceptable level, that is, a level in which sufficient detail is provided to show that the PCI-E interface under test is within a compliant range. This was achieved by degrading the connection to the interface such that it was no longer compliant to the standard. To degrade the signal a series of riser/extension cables were daisy-chained to the interface until the signal was showing high levels of noise and jitter.

To demonstrate the functionality of the designed system, these tests used known good interfaces as a comparison. The comparison interfaces used were two commercially available motherboard PCI-E expansion slots, which were selected as they had previously undergone full compliance testing during their design and manufacture. A number of configurations of riser/extension cables were chosen to gradually degrade the signal integrity. To test if compliance still held, a desktop graphics card was connected to the interface after each scan was completed to see if it would still output a display. As the graphics card requires a x16 connection the USB riser had to be used throughout testing. This riser only passes through 1 lane of the PCI-E and therefore allowed the tests to focus on the degradation of this single lane. The specifics of the configurations are listed below.

- **Configuration 1:** Direct Connection
- **Configuration 2:** USB Riser
- **Configuration 3:** Straight Connector → USB Riser
- **Configuration 4:** Straight Connector → Angle Connector → USB Riser
- **Configuration 5:** Straight Connector → Angle Connector → Angle Piece → USB

For each of the configurations the waveform and BER views were measured and recorded. Then, for an overview of the effect of each configuration the measured parameters were plotted against the configuration number. These were the configurations found to degrade the signal to non-compliance. The extensions and risers used are pictured in Figures 4.1 to 4.4 below.

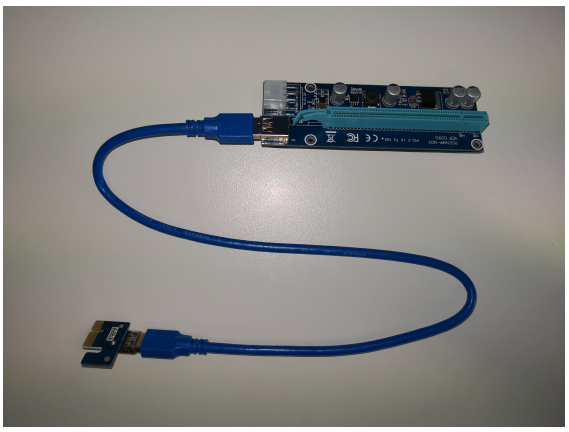

Figure 4.1: USB Riser

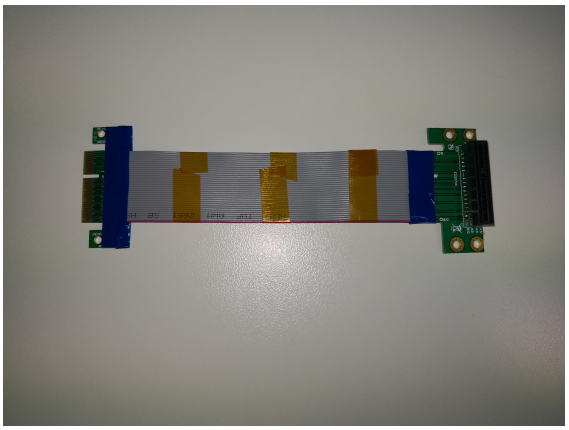

Figure 4.2: Angle Connector

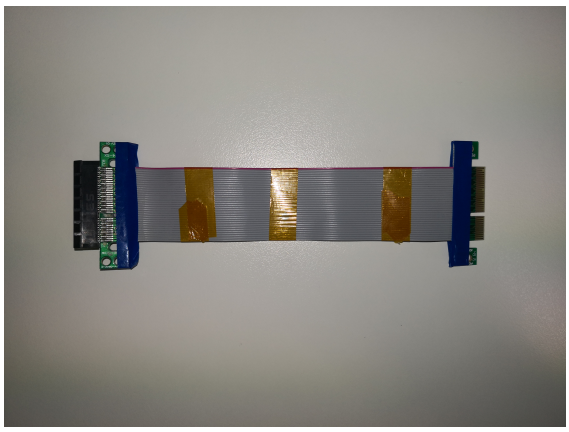

Figure 4.3: Straight Connector

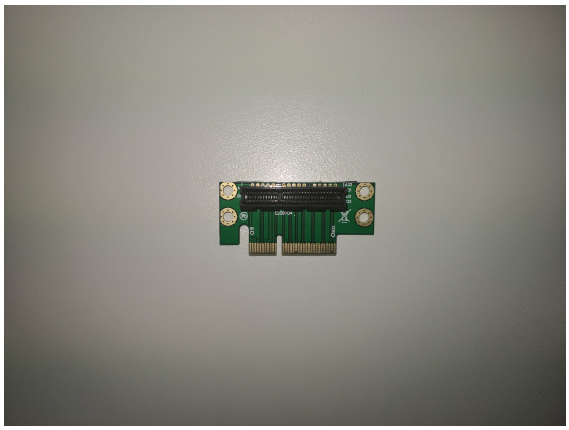

Figure 4.4: Angle Piece

4.1.2 Asynchronous Test Methodology

There were two main stages for the development of the asynchronous eye reconstruction. The first stage was a MATLAB based proof of concept, and the second stage was the same process implemented for use with an Analog Discovery 2 (AD2). The AD2 is a USB oscilloscope, logic analyser, and pattern generator. These were chosen for use as they were readily available and familiar, and therefore suited then needs of testing the implementation. The goal of testing both of these approaches was to prove that the algorithm would work in both a simulation and in a physical implementation.

To generate a test signal for the MATLAB simulation, functions from the communications

toolbox were used. These functions are able to generate a pseudo-random bit sequence with a certain amount of jitter which is then transmitted over a lossy communications channel. This was used as it closely emulates real data, minimising the potential for future challenges when testing with real data. A specified number of data samples were generated per symbol, this data was then down sampled by taking every *nth* sample, where n was an integer larger than the number of samples per symbol. This n was chosen specifically so that it was well below Nyquist sampling rate and would ensure asynchronous undersampling. The code used for this sample generation is detailed in Appendix A.2. An example of the data generated showing the selected undersampled data is shown below in Figure 4.5, where the blue plot is the high frequency data and the orange stems are the undersamples of that data.

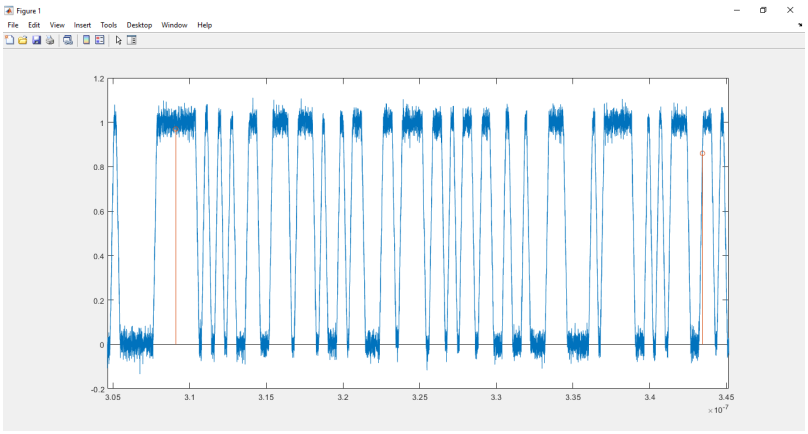

Figure 4.5: Example Data Generated in MATLAB

For the real data the wave generation built into the AD2 was used. This is able to generate a random bit sequence at a specified frequency. This generated output was then connected to the input of another AD2 in the configuration seen below in Figure 4.6.

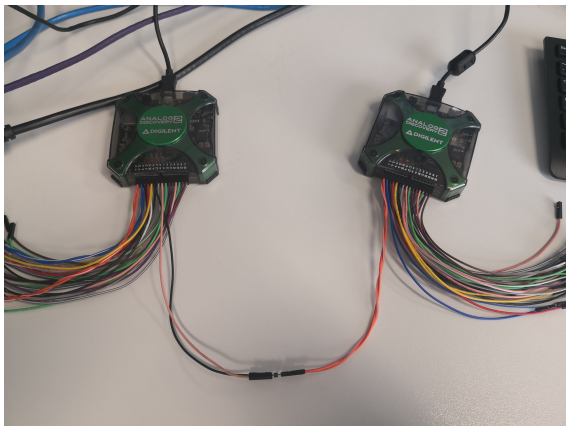

Figure 4.6: Example Data Generated in MATLAB

This second device was set to sample the incoming signal using the decimation sampling configuration. This was chosen as the default setting of the AD2 is to take the average of every 10 samples and given the high speeds of the incoming signals this often led to the samples showing the DC average of the signal. The decimation setting on the other hand only keeps one of these samples, though it is not specified if there is a condition on this or if the choice is random. Regardless, an example of the sampling configuration can be seen in the results images in Section 4.3.

4.2 PCI-E Interface Test Results

4.2.1 ASUS P6T Motherboard

Configuration 1 - Results

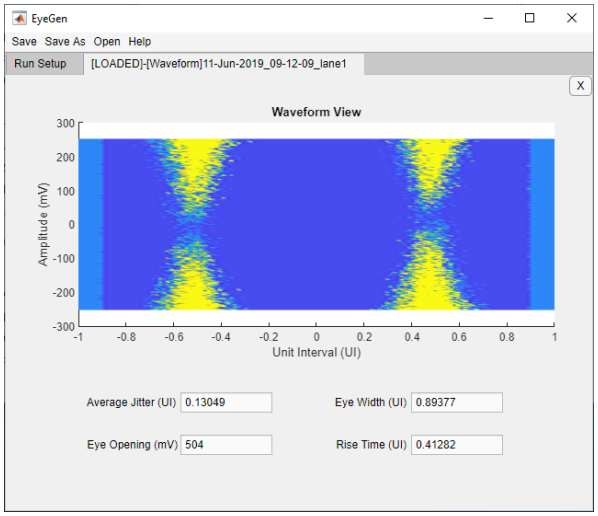

Figure 4.7: Waveform - Configuration 1

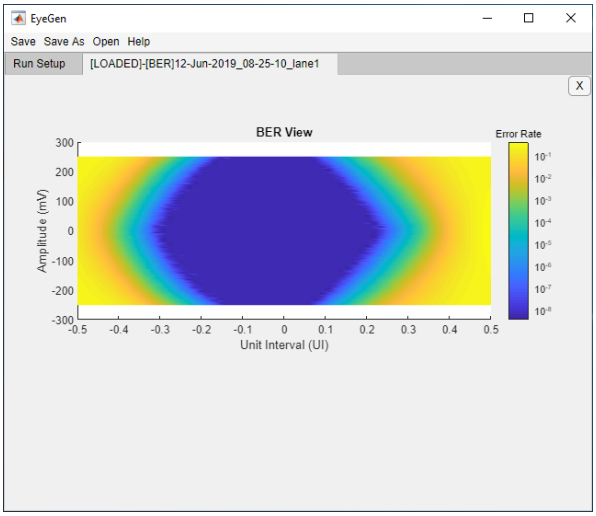

Figure 4.8: BER - Configuration 1

Configuration 2 - Results

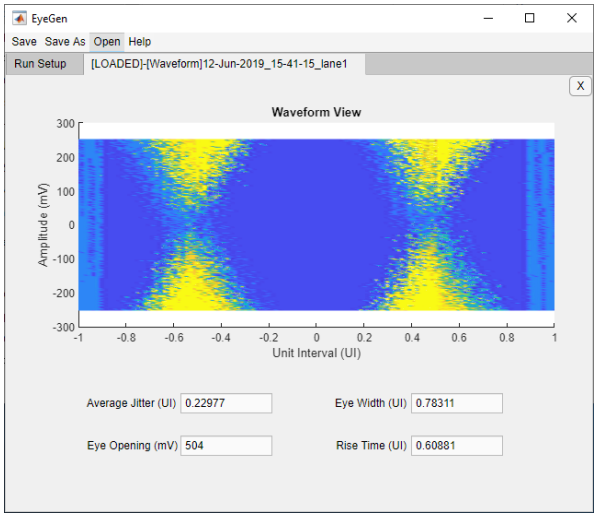

Figure 4.9: Waveform - Configuration 2

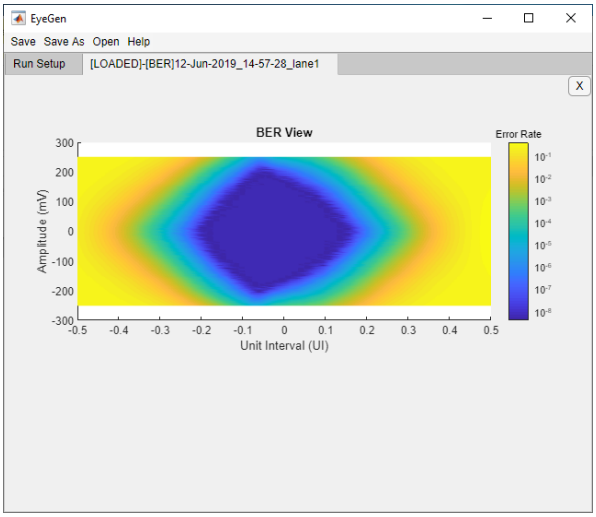

Figure 4.10: BER - Configuration 2

Configuration 3 - Results

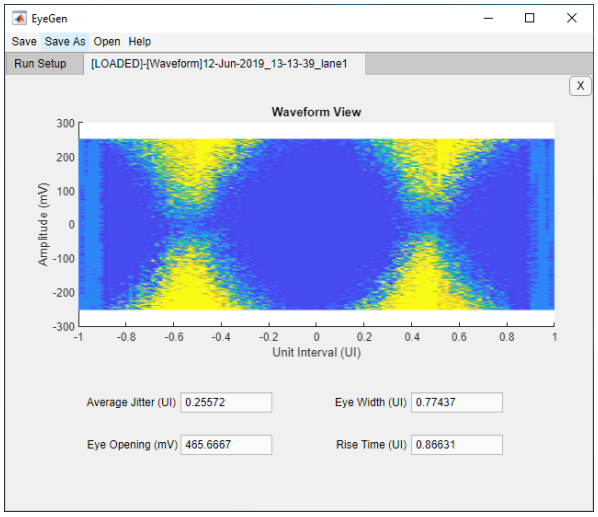

Figure 4.11: Waveform - Configuration 3

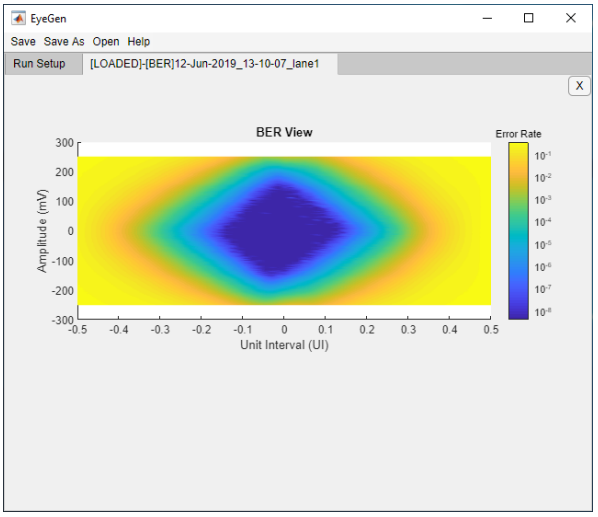

Figure 4.12: BER - Configuration 3

Configuration 4 - Results

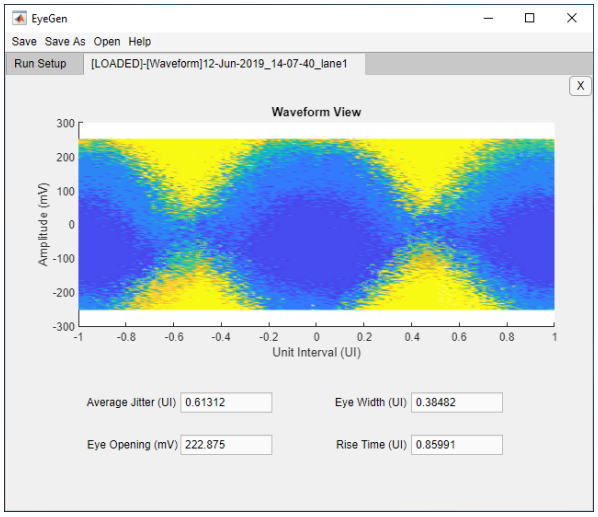

Figure 4.13: Waveform - Configuration 4

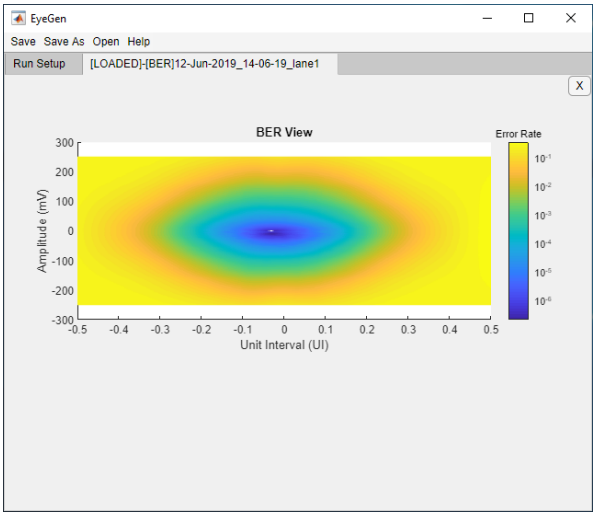

Figure 4.14: BER - Configuration 4

Configuration	Compliance	Graphics Display
1	PASS	PASS
2	PASS	PASS
3	PASS	PASS
4	FAIL	FAIL

Table 4.1: Results on ASUS Motherboard

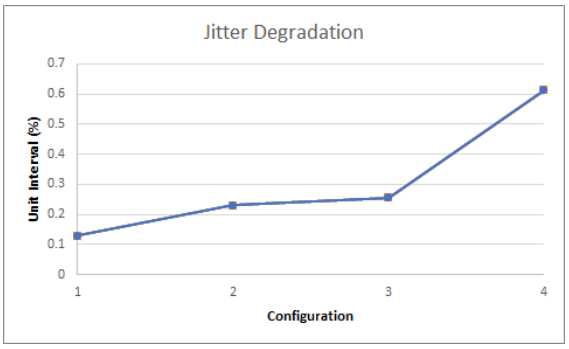

Figure 4.15: ASUS - Jitter Degradation

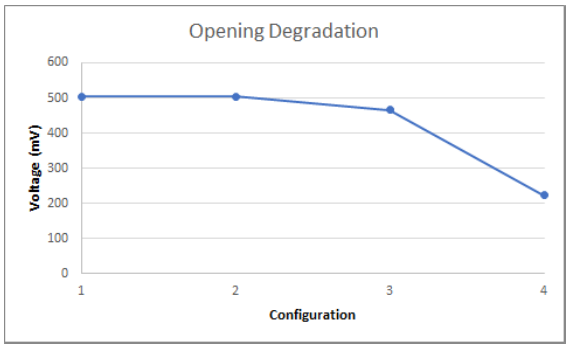

Figure 4.16: ASUS - Opening Degradation

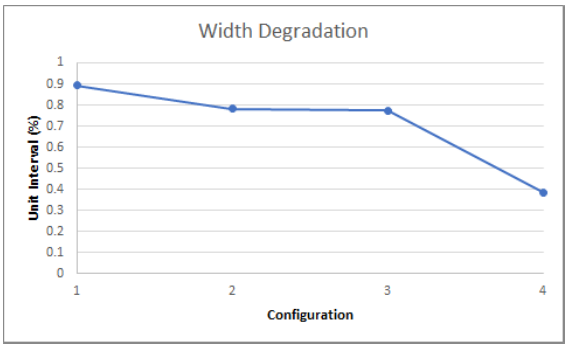

Figure 4.17: ASUS - Width Degradation

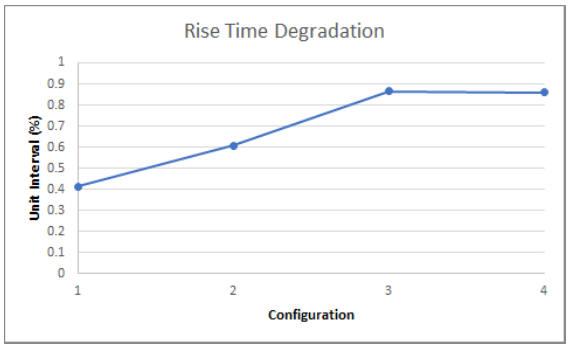

Figure 4.18: ASUS - Rise Time Degradation

4.2.2 Lenovo 03T8244 Motherboard

Configuration 1 - Results

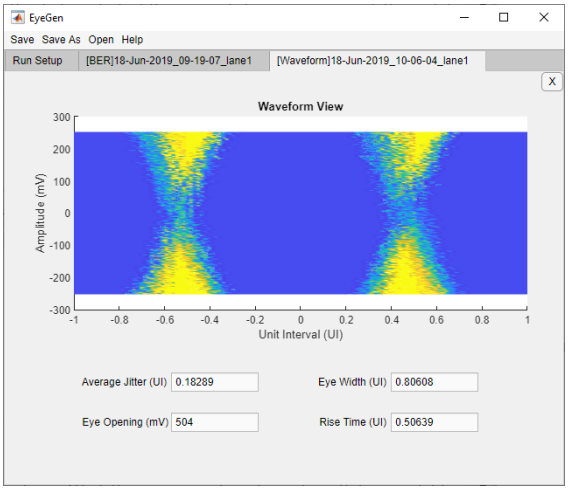

Figure 4.19: Waveform - Configuration 1

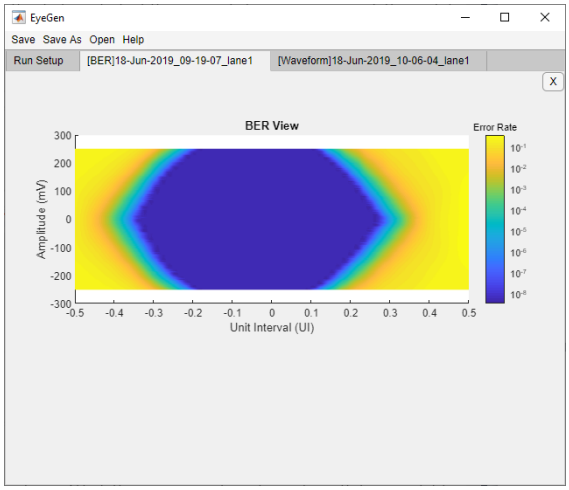

Figure 4.20: BER - Configuration 1

Configuration 2 - Results

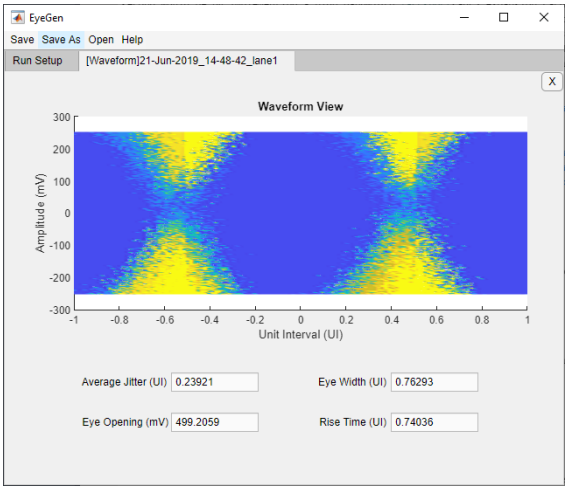

Figure 4.21: Waveform - Configuration 2

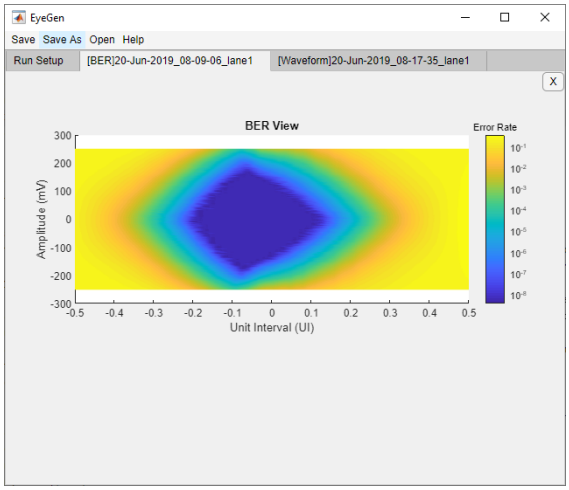

Figure 4.22: BER - Configuration 2

Configuration 3 - Results

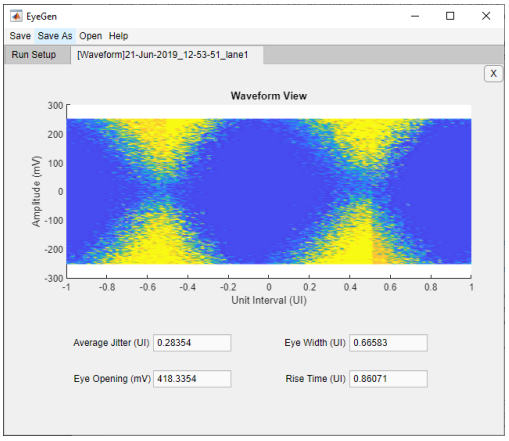

Figure 4.23: Waveform - Configuration 3

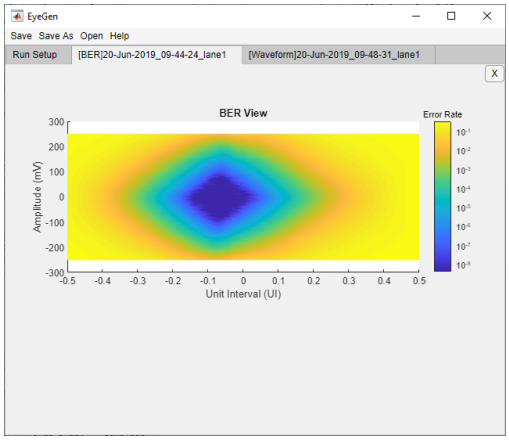

Figure 4.24: BER - Configuration 3

Configuration 4 - Results

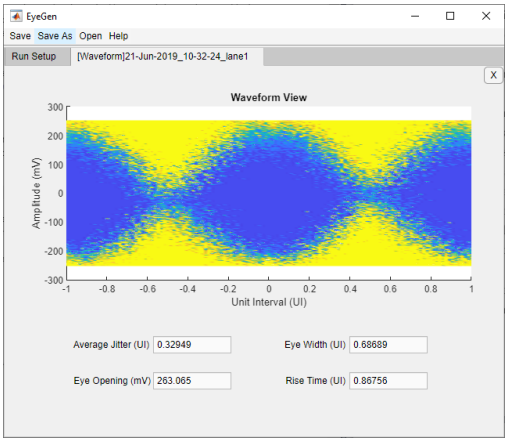

Figure 4.25: Waveform - Configuration 4

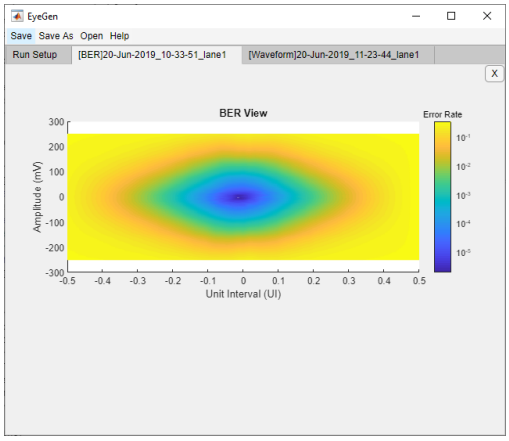

Figure 4.26: BER - Configuration 4

Configuration 5 - Results

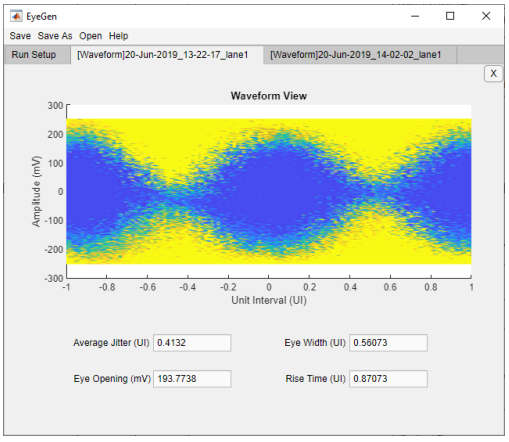

Figure 4.27: Waveform - Configuration 5

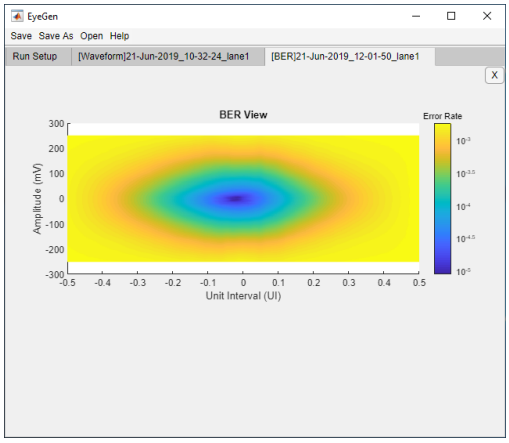

Figure 4.28: BER - Configuration 5

Configuration	Compliance	Graphics Display
1	PASS	PASS
2	PASS	PASS
3	PASS	PASS
4	FAIL	PASS
5	FAIL	FAIL

Table 4.2: Results on ASUS Motherboard

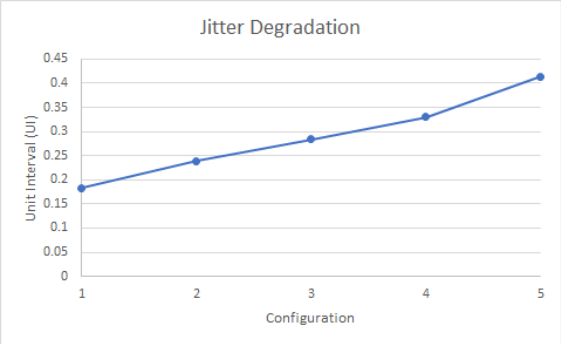

Figure 4.29: Lenovo - Jitter Degradation

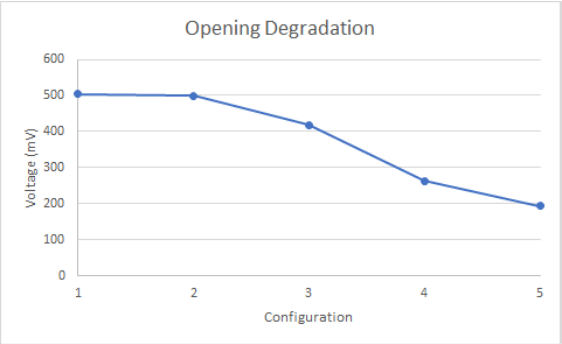

Figure 4.30: Lenovo - Opening Degradation

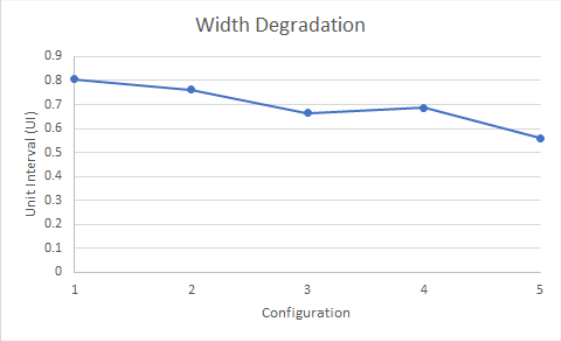

Figure 4.31: Lenovo - Width Degradation

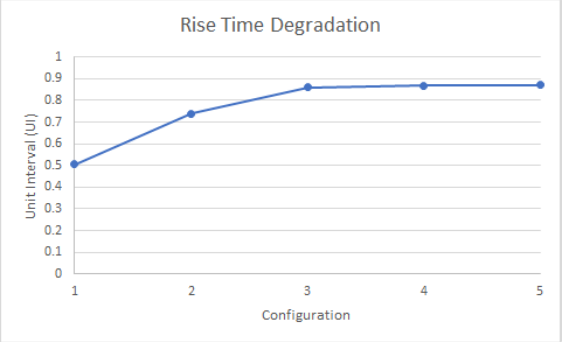

Figure 4.32: Lenovo - Rise Time Degradation

4.3 Asynchronous Test Results

4.3.1 100kHz Signal

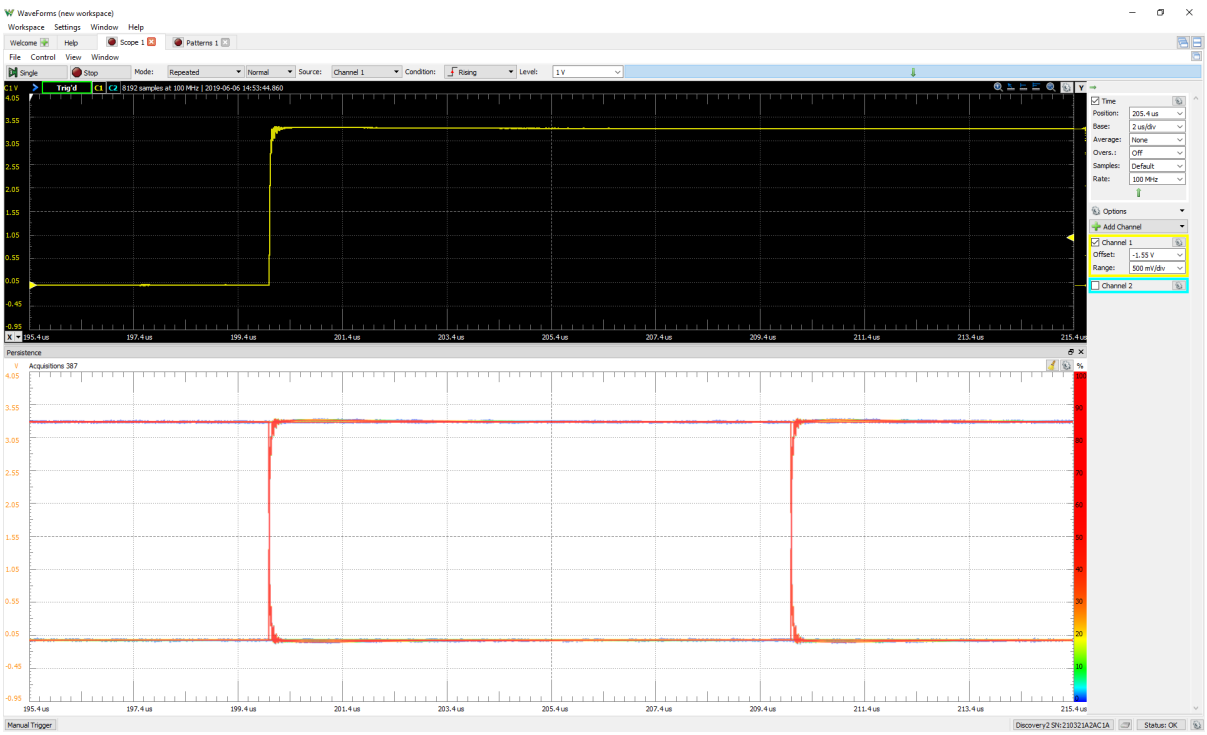

Figure 4.33: 100kHz Signal Sampled at 100Mhz

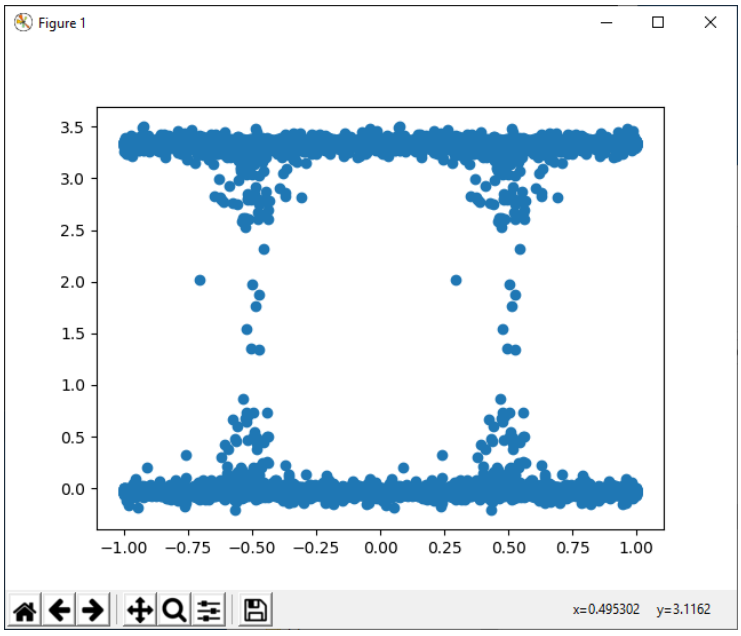

Figure 4.34: Top: Reconstructed 100kHz Signal Sampled at 1643.17Hz

4.3.2 25MHz Signal

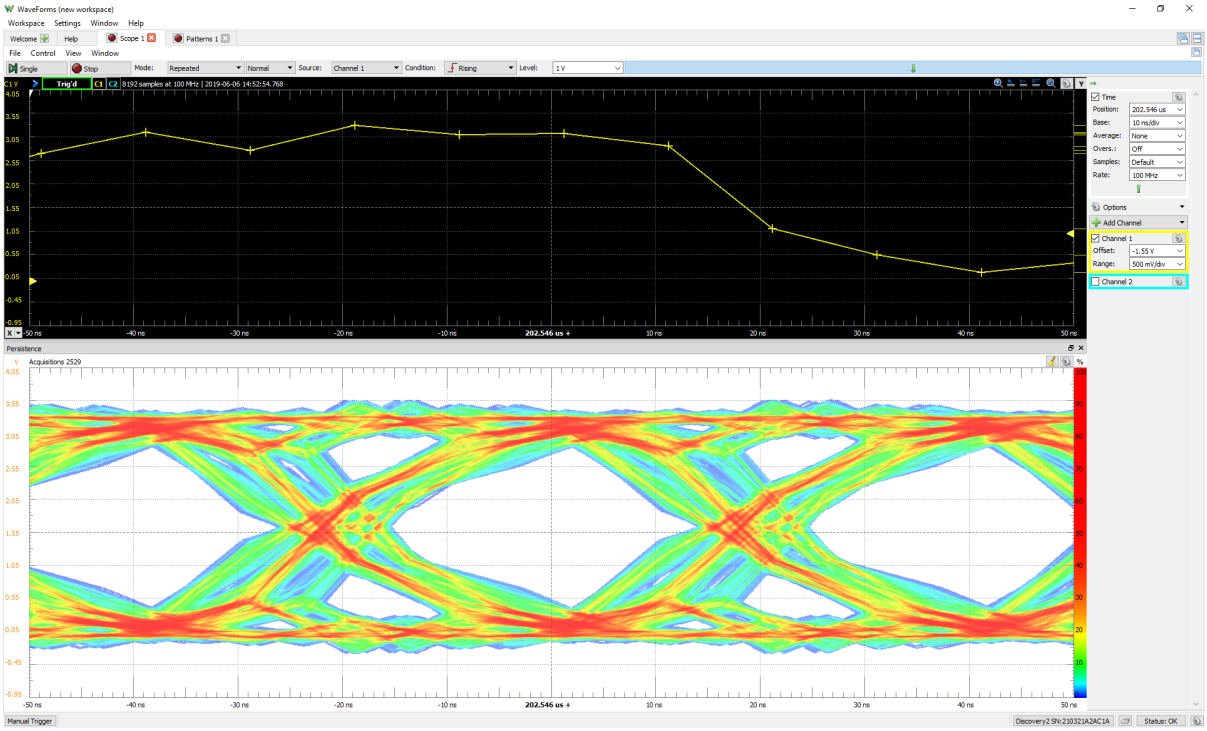

Figure 4.35: 25MHz Signal Sampled at 100Mhz

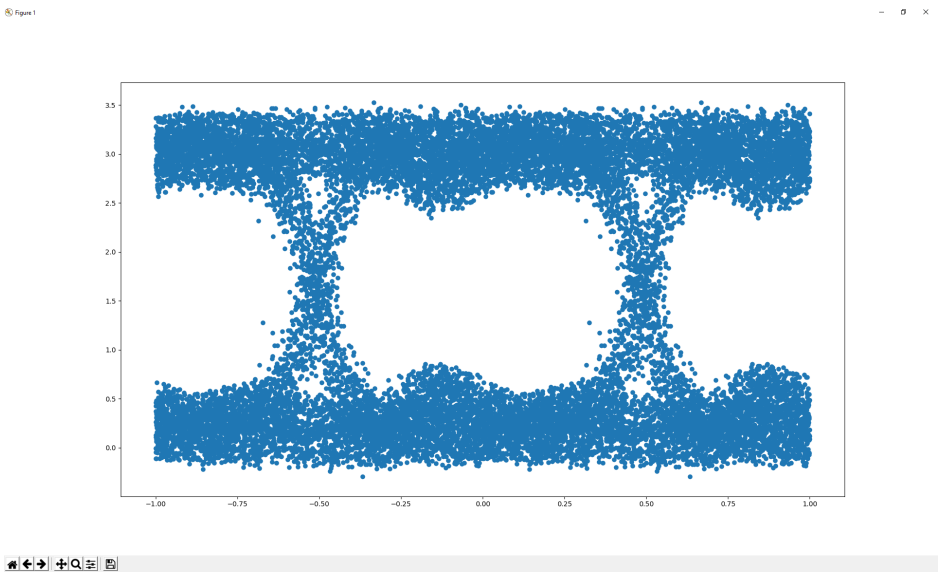

Figure 4.36: Reconstructed 25MHz Signal Sampled at 1643.17Hz

4.3.3 50MHz Signal

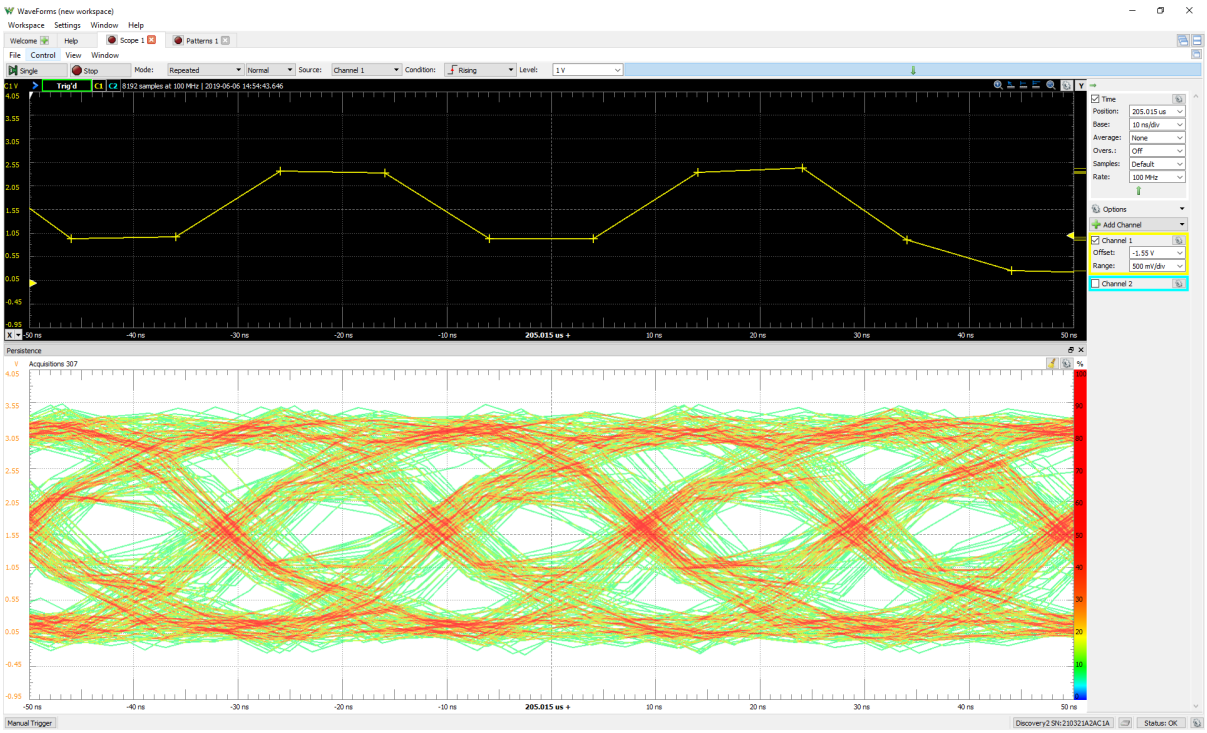

Figure 4.37: 50MHz Signal Sampled at 100Mhz

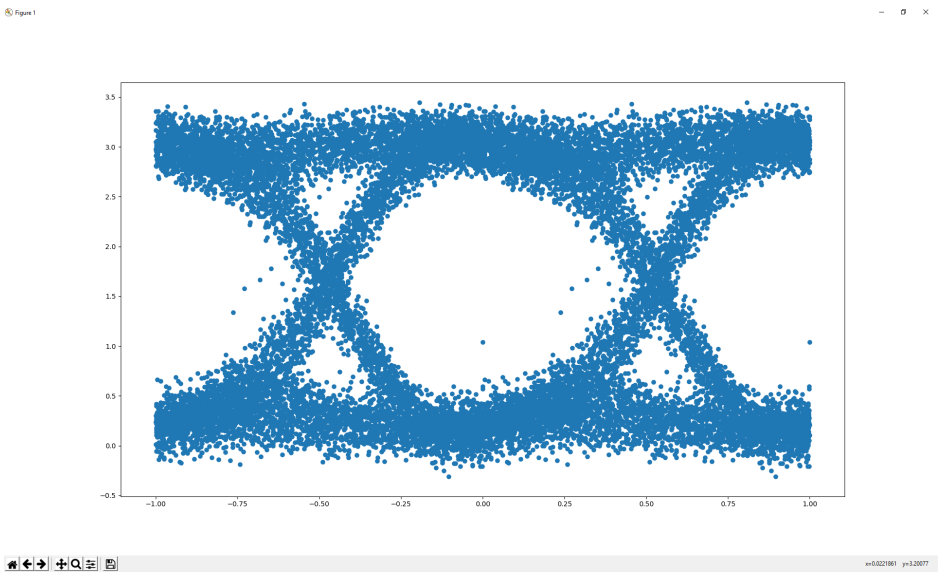

Figure 4.38: Reconstructed 50MHz Signal Sampled at 1643.17Hz

4.3.4 100MHz Signal

Sampling frequency was insufficient for AD2 to reconstruct eye diagram.

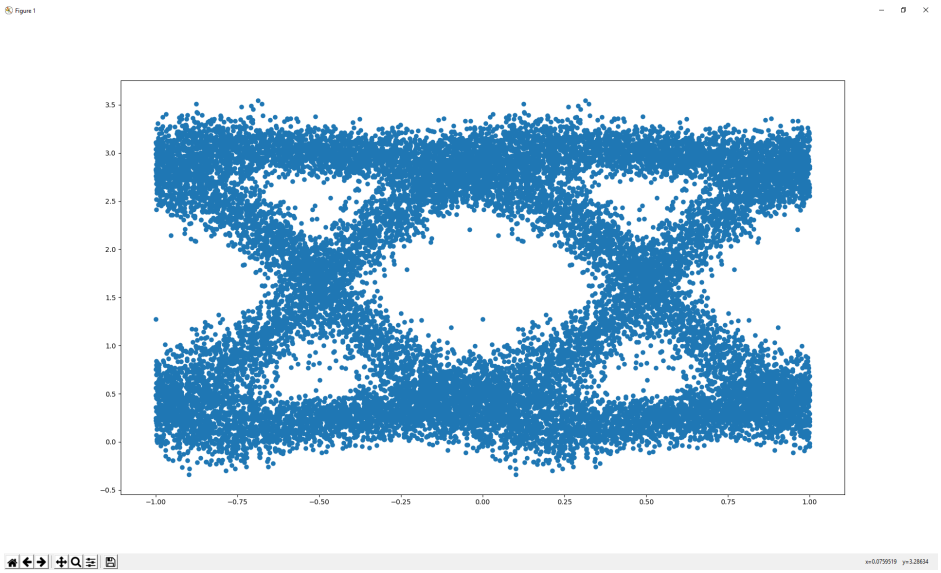

Figure 4.39: Reconstructed 100MHz Signal Sampled at 1643.17Hz

5 Discussion and Recommendations

5.1 Discussion of Results

This section will look to analyse and discuss the results attained in the previous chapter. This discussion will be broken into two parts: the results from the PCI-E scan system, and the asynchronous eye reconstruction system. The goal of collecting these results was to provide confidence in the outputs of each system by testing them on known good systems, or by comparing them with commercial products. It was hoped that confidence could be reached without the need for comparison against known expensive interface testing methods as this would be costly and in direct opposition to the goals of the project as a whole.

5.1.1 PCI-E Scan System

As discussed earlier in the methodology section, the goal was to degrade the signal being transmitted over an interface that had already undergone compliance testing. Each degradation would be followed by connecting a graphics card which would also have undergone compliance testing, with an expectation that the scanned interfaces would show a failure of the graphics card when the scan system registered that there was a non-compliant parameter. The first set of results attained met this expectation well. However, there is significant uncertainty in the measurements for configuration 4 of the ASUS motherboard, which can be seen in the "shadowing" effect in Figure 4.13, though this configurations non-compliance was confirmed when the graphics card was connected. The second set of results attained also met this expectation. However, the test of configuration 4 gives a jitter result that would fail compliance. Though as mentioned previously, there is some uncertainty in the measurement and this result is within 6% of the compliance value and it is possible that the interface may still be able to operate within a small margin of non-compliance. Configuration 5 shows that this system still functions as intended, as a much larger jitter value was measured and as a result the graphics card failed to display as expected. It should be noted that there are vertical bar artefacts in a number of the scan results and, as mentioned previously, the origin of these is unknown but their presence is compensated for by the program.
In both sets of tests the BER is shown alongside the waveform view. This was to gain a qualitative understanding of what this view looks like when the interface fails. In both

test cases it is seen that when the interface fails the eye is almost entirely closed. This confirms that it remains important to complete a BER scan as it provides more understanding about the integrity of the interface.
Overall, this system does give useful information in the validation of the PCI-E interfaces. As originally intended this system was designed to give the engineering team useful information about the performance of the interfaces designed into Opengear's products. Even if this solution is not expanded upon in the future it could reduce verification time for the PCI-E interface during design.

5.1.2 Asynchronous Eye Scan

These results aimed to show that there was a similarity between the eye diagram measured by the AD2 and the one reconstructed by the algorithm. In this case the AD2 was taken to be the known good system as it has an in-built method for recreating an eye diagram. It was expected that the results from the asynchronous reconstruction would have similar features to the eye displayed through the AD2. It was also expected that the eye would begin to degrade on the AD2 as the sampling frequency became insufficient while the asynchronous eye would see no similar degradation. In the first two sets of results shown in Subsections 4.3.1 and 4.3.2 it is seen that the features of the eyes are very similar. In section 4.3.2 a ripple can be observed on the left hand side of AD2 eye, which is reflected in the asynchronous eye, though it is hard to observe due to the differences in horizontal scaling. The result in Subsection 4.3.3 shows the beginning of the degradation of the eye being measured by the AD2, while the asynchronous eye is still very clear. Finally, in Subsection 4.3.4 the AD2 could no longer recover an eye while the reconstruction method displays a result consistent with those previously shown. Overall, these results are consistent with those of the literature as it was able to accurately reconstruct the eye diagram through a high degree of sub-sampling. This proof of concept shows good evidence that this system could be implemented in possible future work.

5.2 Directions for Future Work

There are two main directions that have been foreseen for future work as a result of this project. The first is expanding the PCI-E scanning system to scan other interfaces using similar methods. The second is to implement a high bandwidth ADC expansion card for the FPGA with the asynchronous reconstruction algorithm referred to in Section 5.1.2 to integrate with the original scan system. These two ideas for future work seek to improve upon the original goals of Opengear, which were to test and validate their high speed

interfaces. This project focused specifically on the validation of the PCI-E interface but has laid the groundwork for an expansion to other interfaces.

5.2.1 Expanded Interface Scanning

In order to expand the scanning system to scan other interfaces, the remaining transceivers would be implemented using the same methods described in Section 3.1. The remaining transceivers of the FPGA can be configured to run through the general purpose SMA connectors, which could be used to drive or receive data from a breakout board for other types of interfaces, such as USB or the optical interface. Given the low resource utilisation of this project this could be implemented as an entirely separate system from original scanning architecture. This would have to be achieved by implementing an instance of the interface and sending information over it in the same way as the PCI-E scan system. If there were no IP blocks available the primary obstacle would be implementing instances of an interface in the FPGA logic. This becomes especially problematic for USB, as this is commonly implemented as a separate chip on-board rather than in FPGA logic, and these chips do not possess the secondary samplers required for an eyescan. Though this may be unnecessary if an investigation were conducted into probing an interface that was in use and analysing the raw data that way rather than having the FPGA stimulate the interface directly. Regardless, there are a number of avenues available in this direction and research could be conducted to determine which would be the most suitable.

5.2.2 Asynchronous Eye Recovery Integration

Another possible avenue for future work is to integrate the asynchronous recovery system into the existing scan system. This could be achieved with an attachment card connected to the mezzanine card interface on the AC701 board. This attachment card would need to provide access to at least one high bandwidth ADC that could sample the various high speed signals. These signals would need to be collected by probing an interface in use, as the algorithm requires the interface to be used at 100% capacity. The control of this scanning system could be developed as part of the existing FPGA architecture or as a separate subsystem that communicates back to a PC. It would be advantageous to incorporate the processing software with the software that was developed for this project so that all of the integrity testing could be performed by a single program.

5.3 Project Outcomes

Reflecting on the scope and requirements of this project that were outlined in Section 1.3, both of the designed systems have produced their desired outcomes. In successfully completing its goals this project has provided real value to Opengear. It has delivered a system that is able to test and validate the PCI-E interface, one of the important interfaces within Opengear's devices. It has also provided a proof of concept for a more comprehensive system that could be implemented in the future.

The original primary scope of this project was to collect the information required to construct an eye diagram, such that at least one interface could be quickly and easily characterised and validated. The PCI-E scanning system that was implemented has been demonstrated to achieve this goal. It is able collect enough information to construct both a BER and a waveform eye diagram. The waveform view provides important quantitative measures that characterise and validate the interface being scanned. Similarly, the BER view is able to provide a more qualitative view of the overall compliance of the interface. As a whole, this system is able to produce information that will aid in the design process as it can show with reasonable confidence when an interface is performing in a sub-standard way. This could remove the need for extensive troubleshooting if there were an issue with the design of one of these interfaces.

The final stretch goal of the project was to implement a proof of concept of the asynchronous eye reconstruction algorithm found in literature. As it was demonstrated previously this goal was successfully achieved in MATLAB. This was tested using a simulated non-ideal signal to validate that it would operate correctly in a physical implementation. Time permitted for a physical implementation to be tested by gathering information from a pattern generator using an oscilloscope. This system was able to produce a consistent and accurate result across a range of frequencies and at a high sub-sampling factor. This system in its current configuration could test some of the simpler interfaces such as I^2C or SPI if they were configured to transmit data constantly. There was insufficient time to implement these tests as part of this project, but the pseudo-random bit sequence testing verifies that the system would work in these conditions. This provides a foundation for possible future future work in Opengear if it proves desirable to expand upon the outcomes of this project.

Ideally, both systems would be verified against a known compliance test output. This would further increase the confidence in the outputs of both systems. However, as mentioned throughout the progress of the project and this report, Opengear preferred to avoid unnecessary expense. Both of the designed systems have been implemented using equipment that was already available within Opengear and has incurred no extra expense.

Additionally, the testing that has been conducted has produced systems that have been validated with an acceptable level of confidence while needing no new equipment purchase. Overall, the delivered systems have provided value for Opengear as they are ready to be used and if desired, expanded upon.

This project was a significant undertaking personally, professionally, and academically, and as a result significant challenges were faced in the process of development. These challenges included the initial planning and scoping of the project, contacting academics from other countries about work relevant to the project, and presenting findings in both an academic and professional setting. This project has resulted in a significant amount of personal, academic, and professional development, which is further detailed in the project reflections, found in Appendix C.

References

[1] "Global printed circuit board (pcb) market 2012-2017 & 2018-2023 - growing demand for high speed data and signal transmission, and development of green pcbs," *PR Newswire*, 2018.

[2] *Pcie*, [Accessed: 4-3-19], University of New Hampshire InterOperability Laboratory, 2019. [Online]. Available: https://www.iol.unh.edu/sites/default/files/brochures/PCIe-Brochure.pdf.

[3] *Standards compliance and certification*, [Accessed: 4-3-19], Granite River Labs, 2019. [Online]. Available: https://graniteriverlabs.com/standards-compliance-certification/.

[4] *Test and certification*, [Accessed: 4-3-19], Allion, 2019. [Online]. Available: https://www.allion.com/logo-certification/.

[5] *Compliance program*, [Accessed: 4-3-19], PCI-SIG, 2019. [Online]. Available: https://pcisig.com/developers/compliance-program.

[6] *Compliance*, [Accessed: 4-3-19], USB-IF, 2019. [Online]. Available: https://www.usb.org/compliance.

[7] *N5393f pci express® electrical performance validation and compliance software*, [Accessed: 4-3-19], Keysight, 2018. [Online]. Available: https://www.keysight.com/en/pd-2777519-pn-N5393F/pci-express-electrical-performance-validation-and-compliance-software?pm=OP&nid=-32976.1200321&cc=AU&lc=eng.

[8] *Pcie compliance testing*, [Accessed: 4-3-19], Teledyne Lecroy, 2019. [Online]. Available: https://teledynelecroy.com/protocolanalyzer/pci-express/pcie-compliance-testing.

[9] *Pci express*, [Accessed: 4-3-19], Tektronix, 2019. [Online]. Available: https://www.tek.com/pci-express.

[10] *I2c-bus specification and user manual revision 6*, [Accessed: 22-4-19], Philips Semiconductors, 2014. [Online]. Available: https://www.nxp.com/docs/en/user-guide/UM10204.pdf.

[11] *Pci express base specification revision 4.0*, 2017.

[12] *Pci local bus specification revision 2.2*, 1998.

[13] *Universal serial bus 3.0 specification*, 2008.

[14] *Serial ata revision 3.0*, 2009.

[15] *Commercial building telecommunications cabling standard*, [Accessed: 22-4-19], Telecommunications Industry Association, 2001. [Online]. Available: `https://www.nag.ru/goodies/tia/TIA-EIA-568-B.1.pdf`.

[16] J. Z. Manny Soltero and C. Cockril, *Rs-422 and rs-485 standards overview and system configurations*, [Accessed: 22-4-19], Texas Instruments, 2010. [Online]. Available: `http://www.ti.com/lit/an/slla070d/slla070d.pdf`.

[17] *Pci express base specification revision 1.1*, 2003.

[18] *Universal serial bus 1.0 specification*, 1996.

[19] *Pci express base specification revision 3.0*, 2010.

[20] D. A. Ravi Budruk and T. Shanley, *PCI Express System Architecture*. Mind Share, 2003, ISBN: 0321156307.

[21] *Universal serial bus 2.0 specification*, 2000.

[22] *On-the-go and embedded host supplement to the usb revision 2.0 specification*, 2012.

[23] D. Anderson and J. Trodden, *USB 3.0 Technology*. Mind Share, 2013, ISBN: 9780983646518.

[24] *Standard for digitizing waveform recorders*, IEEE Std. 1057:2017, 2017.

[25] *Standard for terminology and test methods for analog-to-digital converters*, IEEE Std. 1241:2010, 2010.

[26] *Standard for transitions, pulses, and related waveforms*, IEEE Std. 181:2011, 2011.

[27] *Definitions and terminology for synchronisation networks*, ITU-T Recommendation G.810(08/96), 1996.

[28] J. Moreira and H. Werkmann, "An engineer's guide to automated testing of high-speed interfaces," in. Artech House Inc, 2010, ch. 3 High Speed Interface Standards, ISBN: 1607839830.

[29] *Understanding data eye diagram methodology for analyzing high speed digital signals*, [Accessed: 4-3-19], ON Semiconductor, 2015. [Online]. Available: `https://www.onsemi.com/pub/Collateral/AND9075-D.PDF`.

[30] J. Moreira, *An engineer's guide to automated testing of high-speed interfaces*, eng, ser. Artech House microwave library. Boston: Artech House, 2010, ISBN: 9781607839835.

[31] Y. Fan and Z. Zilic, *Accelerating Test, Validation and Debug of High Speed Serial Interfaces*. Dordrecht: Springer Netherlands, 2011, ISBN: 9789048193974.

[32] *7 series fgpas gtp transceivers user guide*, [Accessed: 4-3-19], Xilinx, 2016. [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug482_7Series_GTP_Transceivers.pdf.

[33] G. Moustakides, F. Cerou, O. Audouin, and L. Noirie, "Eye diagram reconstruction using asynchronous imperfect sampling, application to ber estimation for fiber-optic communication systems," in *2002 11th European Signal Processing Conference*, Sep. 2002, pp. 1–4.

[34] S. Zheng, "A low cost asynchronous eye diagram reconstruction system for high speed links," Master's thesis, Massachusetts Institute of Technology, 2013.

[35] S. Gupta and A. Phatak, *Adc guide, part 2: The sample rate*, [Accessed: 24-5-19], Cypress Semiconductor Corp., 2012. [Online]. Available: https://m.eet.com/media/1158578/c0895pt2.pdf.

[36] *Application note 775 specifications and architectures of sample-and-hold amplifiers*, [Accessed: 24-5-19], Texas Instruments, 1998. [Online]. Available: http://www.ti.com/lit/an/snoa223/snoa223.pdf.

[37] W. Kester, *Aperture time, aperture jitter, aperture delay time — removing the confusion*, [Accessed: 24-5-19], Analog Devices, 2009. [Online]. Available: https://www.analog.com/media/en/training-seminars/tutorials/MT-007.pdf.

[38] *Ads5400 12-bit, 1-gsps analog-to-digital converter*, [Accessed: 24-5-19], Texas Instruments, 2009. [Online]. Available: http://www.ti.com/lit/ds/symlink/ads5400.pdf.

[39] *Ac701 base targeted reference design user guide*, [Accessed: 22-4-19], Xilinx, 2014. [Online]. Available: https://www.xilinx.com/support/documentation/boards_and_kits/ac701/2015_1/ug964-ac701-trd-ug.pdf.

[40] R. Munden, "The design warrior's guide to fpgas," English, *Printed Circuit Design & Manufacture*, vol. 21, no. 7, 2004, ISSN: 1543-6527.

[41] *Ac701 evaluation board for the artix-7 fpga*, [Accessed: 18-3-19], Xilinx, 2015. [Online]. Available: https://www.xilinx.com/support/documentation/boards_and_kits/ac701/ug952-ac701-a7-eval-bd.pdf.

[42] *7 series fpgas transceivers wizard v3.6*, [Accessed: 10-3-19], Xilinx, 2016. [Online]. Available: https://www.xilinx.com/support/documentation/ip_documentation/gtwizard/v3_6/pg168-gtwizard.pdf.

[43] *Bridging an axi4-lite interface to drp interfaces*, [Accessed: 22-4-19], Xilinx, 2015. [Online]. Available: https://www.xilinx.com/support/documentation/application_notes/xapp1214-drp-bridge.pdf.

[44] *Amba axi and ace protocol specification*, [Accessed: 22-4-19], ARM, 2011. [Online]. Available: https://silver.arm.com/download/download.tm?pv=1198016.

[45] *Axi reference guide*, [Accessed: 22-4-19], Xilinx, 2012. [Online]. Available: https://www.xilinx.com/support/documentation/ip_documentation/axi_ref_guide/latest/ug761_axi_reference_guide.pdf.

[46] *Amba 4 axi4-stream protocol specification*, [Accessed: 22-4-19], ARM, 2010. [Online]. Available: https://silver.arm.com/download/download.tm?pv=1074010.

[47] *Microblaze processor reference guide*, [Accessed: 22-4-19], Xilinx, 2009. [Online]. Available: https://www.xilinx.com/support/documentation/sw_manuals/xilinx11/mb_ref_guide.pdf.

[48] *Set up for matlab axi master*, [Accessed: 22-4-19], MATLAB, 2019. [Online]. Available: https://au.mathworks.com/help/supportpkg/xilinxfpgaboards/ug/set-up-for-matlab-axi-master.html.

[49] J. Gabauer, "Test and validation of the integrity and performance of high speed interfaces: Project proposal," University of Queensland, 2019.

[50] ——, "Test and validation of the integrity and performance of high speed interfaces: Interim report," University of Queensland, 2019.

[51] *In-system eye scan of a pci express link with vivado ip integrator and axi4*, [Accessed: 22-4-19], Xilinx, 2014. [Online]. Available: https://www.xilinx.com/support/documentation/application_notes/xapp1198-eye-scan.pdf.

[52] *Using the memory endpoint test driver (met) with the programmed input/output example design for pci express endpoint cores*, [Accessed: 22-4-19], Xilinx, 2009. [Online]. Available: https://www.xilinx.com/support/documentation/application_notes/xapp1022.pdf.

[53] *Rt-eye® pci express® compliance module methods of implementation (moi)*, [Accessed: 24-4-19], Tektronix, 2005. [Online]. Available: http://www.av.it.pt/medidas/data/Manuais%5C%20&%5C%20Tutoriais/60%5C%20-%5C%20MSO71604C/Optional%5C%20SW/Documents/RT-EYE_Documentation/PCIExpress.MOI.pdf.

[54] *Numpy for matlab users*, [Accessed: 28-5-19], Mathesaurus, 2007. [Online]. Available: http://mathesaurus.sourceforge.net/matlab-numpy.html.

[55] *Numpy for matlab users*, [Accessed: 28-5-19], The SciPy Community, 2019. [Online]. Available: https://docs.scipy.org/doc/numpy/user/numpy-for-matlab-users.html.

Appendix A - Code Segments

A.1 MATLAB Asynchronous Eye Reconstruction

```
function [t, x] = eyeReconstruction(samples)
    % Non-linear transform
    tformd = abs(samples - mean(samples)).^0.2;

    % Calculate the dimensions of the shaped matrix
    N = length(f);
    L = 512;
    M = floor(N/L);

    % Reshape the data into blocks
    shaped = reshape(tformd(1:(L*M)), L, M)';

    % Calculate the periodogram
    P = (sum(abs(fft(shaped, [], 2)).^2)./M);

    % Find the peak frequency
    peaks = sort(findpeaks(P(1:floor(length(P)/2))));
    wp = (find(P == max(peaks))/length(P))*2*pi;
    wp = wp(1);

    % Window Size
    K = 256;

    % Deliberately slow phase correction to show steps
    Tn = zeros(1, N);
    y = zeros(2*K+1, 1);
    wind = blackman(2*K+1);
    for n = 1:N
        for k = -K:K
            if ( ((n+k) < 1) || ((n+k) > N) )
                y(k+K+1) = 0;
            else
                y(k+K+1) = wind(k+K+1) * f(n+k) * exp(-(1i*k*wp));
            end
        end
        Tn(n)= angle(sum(y))/(2*pi);
    end

    % Set function outputs to be plotted
    t = Tn;
    x = samples;
end
```

A.2 MATLAB Signal Generation

```
Rs = 2.5E9;              % Symbol rate
Fs = Rs*100;             % Sample rate
sps = Fs/Rs;             % Number of samples per symbol
Trise = 1/(2*Rs);        % Rise time of the NRZ signal
Tfall = 1/(2*Rs);        % Fall time of the NRZ signal
frameLen = 4000000;      % Number of symbols in a frame
undersampF = 8586;       % Factor used for underesampling

% Configuring a "realistic" pseudo-random bit sequence signal source
src = commsrc.pattern('SamplingFrequency', Fs, ...
                      'SamplesPerSymbol', sps, ...
                      'OutputLevels', [0 1], ...
                      'DataPattern', 'PRBS7', ...
                      'RiseTime', Trise, ...
                      'FallTime', Tfall);

% Adding jitter to the source
jitterSrc = commsrc.combinedjitter('DiracJitter','on', ...
                      'DiracDelta', 0.05/Rs*[-1 1], ...
                      'SamplingFrequency', Fs, ...
                      'RandomJitter', 'on', ...
                      'RandomStd', 0.01/Rs);
src.Jitter = jitterSrc;

% Generating signal
tmp = generate(src,frameLen);

% Passing signal through a noisy comm channel
channel = comm.AWGNChannel('NoiseMethod', 'Signal to noise ratio (SNR)', ...
                'SNR', 30, 'SignalPower', 1);

message = channel(tmp);

undersamples = message(1:usampF:end);
```

Appendix B - Project Plan

B.1 Project Methodology

The core project was planned using the waterfall method, as it allows for the development of a fully fleshed out project plan prior to commencement of work. However, this method has only been used to establish a minimum viable product for the project, to allow time at the end of development for resources to be allocated to the development of potential stretch goal features. This component of the project will rely on a more iterative approach, as these goals are more flexible and may require a review of the structure for their execution as the difficulty and necessity of their implementation are established.

B.2 Project Timeline Details

The following timeline will not relate specifically to the academic requirements of this project, as though they effect the timing of certain milestones within the project they are not crucial to the development of the final product. There are a number of changes at this point due to the addition of a new stretch goal and the lack of availability of certain resources. Additionally, to bring the timeline details into line with the tasks outlined in the Gantt chart the majority of the tasks that follow have been updated. It should also be noted that the required times for some of these tasks have been extended under the condition that they were completed concurrently with others.

Induction and Familiarisation - (Complete)
Time Required: 1 Week
Time spent getting my work space set up and being inducted. Along with familiarising myself with some of the products that Opengear produces to get an idea for the applications that this project could have in the design process.

Research Existing/Possible Solutions - (Complete)
Time Required: 1.5 Weeks
Given the change of scope, an informal annotated bibliography was produced. This was in order to gain an understanding of the possible solutions already available and what might be included in the new project scope.

Investigate Implementations and Scope Project - (Complete)
Time Required: 1 Week
The project scope was left relatively open-ended so a meeting was held to brainstorm ideas for what the final scope of the project should look like. This was followed by a revision and continuation of the investigation of the plausible development avenues and an initial draft of the project scope.

Test Basic/Reference Designs on FPGA - (Complete)
Time Required: 1 Week
A week to probe into some of the technical aspects of the various proposed implementations. Tested some basic implementations in a practical manner to get an understanding of the difficulty of the various parts of the project.

Project Planning Finalisation - (Complete)
Time Required: 1.5 Weeks
First draft of the project plan with a clear layout of what the minimum viable product will be and lay out a series of stretch goals that will be attempted to be implemented given enough time.

AXI Implementation - (Complete)
Time Required: 2 Weeks
MATLAB has the capability to interface with an FPGA directly through an AXI interface which allows easier access to the data collected by the interfaces. This is implemented through an IP block provided by MATLAB that will need to be correctly configured and tested to allow the project to proceed.

Program MicroBlaze - (Complete)
Time Required: 3 Weeks
The Microblaze reference design needed to be modified to include a second mode for the waveform eye diagram scan and transmit this data. It also needed to then be able to reset itself into a neutral state so that it could scan again, possibly in a different mode.

Write MATLAB Backend Processing - (Complete)
Time Required: 4 Weeks
Collecting and processing the data that was collected from the FPGA. The final product of this code was processed data that was able to be displayed as an eye-like diagram by MATLAB. The various components of this needed to be made modular for the integration into a GUI.

Investigation of PCI-E Loopback Testing Interface - (Complete)
Time Required: 1 Week
An initial investigation and attempt to implement a less cumbersome PCI-E interface that would allow for testing at higher speeds than the reference design.

MATLAB GUI Development and Polish - (Complete)
Time Required: 4 Weeks
A GUI is currently under development that supports the two different scan modes that have been implemented and the display of their results. It also supports the saving and loading of data from previous scans.

Interface Testing and Bug Fixing - (Complete)
Time Required: 1 Week
Collect results of different PCI-E interfaces to build up a image of what good and bad PCI-E interfaces look like. This will be conducted across a number of devices. Also to act as time for fixing any bugs that appear during this data collection.

Asynchronous Proof of Concept in MATLAB - (Complete)
Time Required: 2.5 Weeks
Implementation of the algorithm found into MATLAB using dummy data to ascertain the difficulty of a fully integrated implementation. Allowing time to resolve any issues with the overall implementation before using the physical hardware.

Interface AMS101 ADC Card
Time Required: 1 Day - (Complete)
Interfaced with desired ADC daughter card through a reference design. Through this initial process realised this would not produce the desired results and an alternative was found.

Implement Asynchronous Eye Recovery - (Complete)
Time Required: 1.5 Weeks
An alternative physical interface was found in an AD2 and a script was written to automatically collect and display the reconstructed eye in Python using the AD2's API.

Bug Fix Eye Scan Algorithm - (Complete)
Time Required: 1 Week
Testing on physical interfaces and comparison against known results or compliance standards to measure the success of the implementation. This was done entirely through the AD2.

Final Interface Testing - (Complete)

Time Required: 1 Week

Final collection of interface data for use in report. Redid some of the original data to get higher resolution scans.

Allowed Slack - (Complete)

Time Allowed: 1 Week

Extra time to allow for academic requirements to be fulfilled, tasks that run long, and other possible time loss. In the end this was mostly consumed by various illnesses and at the end of the project this remaining week was used for academic writing.

B.3 Project Gantt Chart

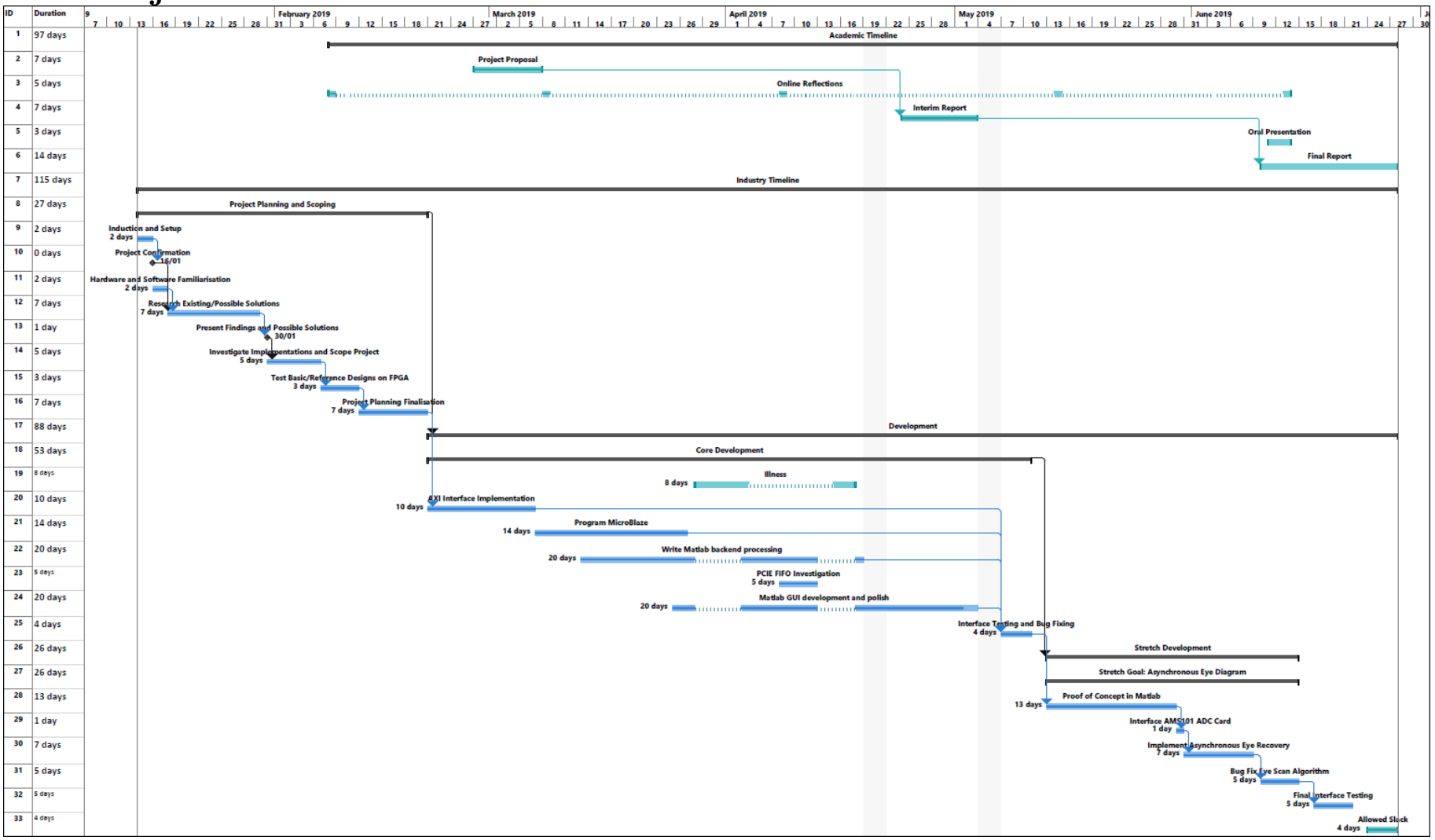

Figure B.1: Final Project Timeline

B.4 Resources

There are a number of resources that will be needed to successfully complete this project. These resources can be categorised into software and hardware and then further categorised into what is currently available and unavailable.

- **MATLAB:** This will be used for the development of the GUI and for the proof of concept of the eye reconstruction algorithm. MATLAB will also provide the AXI interface for communications between the PC and FPGA.
- **Vivado:** This integrated development environment (IDE) is used for the development of code and programming of the FPGA, and the MicroBlaze
- **AC701 FPGA Development Board:** This is the FPGA board that will be programmed for the main functions of the project and provides the main interface that will be used for testing.
- **Analog Discovery 2:** Two of these devices were acquired to test the asynchronous eye reconstruction algorithm. One to generate patterns and one to sample them.

B.5 Risk Assessment

In order to correctly assess and categorise the potential risks to the project, matrices defining risk likelihood, consequence, and residual risk rating have been developed. For each residual risk rating a recommended minimum level of intervention has been specified with each of these recommendations. The risks were revised over the course of the project. New risks did present themselves while others were no longer relevant, however, this this did not change any of the definitions.

B.5.1 Risk Likelihood Definitions

The following table will define the levels of likelihood and frequency of occurrence

Likelihood	Occurrence
Very Likely - (1)	Hazard or risk is likely to occur once every day or two
Likely - (2)	Hazard or risk is likely to occur once per week or fortnight
Possible - (3)	Hazard or risk is likely to occur once per month or two
Unlikely - (4)	Hazard or risk is likely to occur once in the lifetime of the project
Very Unlikely - (5)	Hazard or risk is unlikely to occur in the lifetime of the project

Table B.1: Definitions of Likelihood of Risk Occurrence

B.5.2 Risk Consequence Definitions

The following table establishes contextual conditions to each of the rated risk levels. The categories of risk that have been considered for this project are those relating to health and safety, and scheduling interruptions.

Rating	**Consequence**	
	Health & Safety	Scheduling Interruptions
Catastrophic - (1)	Fatality	Extensive delay resulting in incomplete project
Major - (2)	Permanent Disability	Delays resulting in significant reduction of scope
Moderate - (3)	Hospitalisation	Delay that may require a review of scope
Minor - (4)	Medical Treatment	Delay with no long term effect
Insignificant - (5)	First Aid	Unnoticeable delay

Table B.2: Definitions of Risk Consequence Severity

B.5.3 Risk Matrix

To establish levels of acceptable risk as well as an overall categorization and priority to compare risks with each other the risk matrix below was implemented. The level of risk was assigned by the product of the likelihood and consequence shown in table B.3.

Consequence	**Likelihood**				
	Very Likely - (1)	Likely - (2)	Possible - (3)	Unlikely - (4)	Very Unlikely - (5)
Catastrophic - (1)	1 - Extreme	2 - Extreme	3 - High	4 - High	5 - Moderate
Major - (2)	2 - Extreme	4 - High	6 - Moderate	8 - Moderate	10 - Low
Moderate - (3)	3 - High	6 - Moderate	9 - Moderate	12 - Low	15 - Low
Minor - (4)	4 - High	8 - Moderate	12 - Low	16 - Low	20 - Low
Insignificant - (5)	5 - Moderate	10 - Low	15 - Low	20 - Low	25 - Low

Table B.3: Risk Matrix

B.5.4 Identified Risks

Illness - Low

Impact: An amount of time lost for work depending on the severity of the illness

Control and Mitigation: Continue healthy habits of diet and sleep and if unwell take the necessary time to properly recover.

Ongoing Health Concerns - Moderate
Impact: Lost work time due to appointments and possible days off for recovery
Control and Mitigation: Continue healthy habits of diet and sleep and if unwell take the necessary time to properly recover.

Hardware Failures - Moderate
Impact: Loss of data or disruption of use of hardware
Control and Mitigation: Implement proper source control, and back up any other relevant materials physically and/or online.

Electrocution - Moderate
Impact: Possible need for medical treatment and loss of work time
Control and Mitigation: Not removing shielding from electronics when powered on referring to senior engineers if unsure.

Office work related strain injuries - Low
Impact: An amount of time lost for work depending on the severity of the illness
Control and Mitigation: Regular breaks, and proper support and posture.

Burns - No Longer Applicable
Impact: First aid may be required
Control and Mitigation: Use correct technique and procedures when operating soldering iron.

B.6 Commercial Issues

The intent of this project is not to commercialise the final product. In reality, the final goal is for this product to be used as in-house prototype test equipment. However, if in the future it was decided that this design would move into production there would be some risks involved. Steps would need to be taken to legally protect the production of the product. Currently, this project is largely being developed using MATLAB under an academic license. If the designs were to be moved into production and sold, new licenses would need to be acquired as the current license stipulates that its uses can only be noncommercial and for completion of course requirements. Additionally, if production was to be commenced and FPGAs were continued to be used, then the security of the designs must be maintained as the bitstream containing the configuration for the device may be stored in external memory. If these bitstreams are correctly encrypted then appropriate levels of confidentiality can be maintained.

Appendix C - Professional Development

C.1 Reflection on Placement Learning

This project is the largest single undertaking of my academic career to date. Prior to starting, I had little experience in planning and scoping a project of such magnitude, and purely academic experience in FPGA development. As such, I have faced many challenges over the course of this project and have experienced significant learning and growth.

The first challenge I experienced was a complete change of project in the first week of placement. Due to a miscommunication, I was instead tasked with a project I felt I had little experience with. To remedy this, I conducted a full literature review to educate myself about the project area and to communicate my understanding to Opengear. This meant that I was able to receive tailored feedback and was able to quickly recover my footing. Through this process I also learned how to conduct a formal literature review, which is something I had little experience in previously.

The knowledge gained through the literature review helped the scoping process immeasurably. As the new scope of the project was largely open-ended I had a number of meetings to scope the project. I had to think about where my strengths lie and what I would be able to achieve in the allowed time. I also had to produce something of value for Opengear, which overall made this seem like a daunting task. All of this led me to the project that I have successfully delivered.

There was a significant technical challenge in the implementation of the asynchronous reconstruction algorithm. Initially, this was in understanding the concepts and intent of the algorithm and discerning value in its implementation. This was overcome by following the mathematical reasoning for the algorithm along with reading papers that cited this one as follow-up verifications of the same algorithm. It was thought this was understood but there were troubles with the implementation. It ended up taking twice as long as planned to implement the reconstruction method. I found that I had misunderstood the

nature of the non-linear transform and how to correctly asynchronously undersample a signal. It was suggested to me that I get into contact with some of the Authors of the papers I had read and explain the issues I was having which they helped remedy. I think this was an important lesson because until it was suggested to me I hadn't thought this was an option and I think it is essential to be able to reach out into the wider engineering community.

The situation described above was the first time I had needed to contact an academic about their work. This isn't so much of a challenge in itself but considering the paper was written in 2002 a number of the original academics had retired or no longer had access to the work that was done. Once I did get in contact with a couple of them I had to manage the expectations of the exchange. In one instance I really wanted a quick turnaround time but accepted that I needed to be patient. This ultimately lead to advice that lead me to solve the problems that I was facing. However, in another instance I responded explaining that the problems were resolved but it was requested that I send the code I had been working on anyway. I felt rather hesitant about this had to work around handing over my work. This combination of interactions were rather new to me as in the past when I have requested information it has been in a business sense which has always been quick rather than this case which was asking someone to take their personal time to help me.

Another challenge I have had to face is that I have had little experience working on a large technical project independently. Although, Opengear is a team environment my project is largely unrelated to those outside of my direct team. Given the flexibility of the project I had valued the feedback I had received so far and wished to pull from a wider audience. To achieve this, I gave a short presentation which has helped me to gain new insights from members of the wider Opengear team who I have had limited contact with thus far. Through this process I put to use the skills I learned in ENGG4900 and gave my first formal project presentation in a workplace.

I also utilised this audience that has been available to me to get feedback on the program that I have developed. Once I had the first version of the software I demonstrated it to a number of the engineers in the hardware team.

Finally, as I have mentioned in my monthly reflections, I have experienced numerous health concerns this semester which have cost me a considerable amount of time due

to treatment and recovery. As such, this has necessitated the development and strict adherence to personal planning and time management strategies, which I have not had to use to such a degree in the past. This has taught me a great deal about my own working style, and how best to plan around my productivity cycles.

C.2 Development of EA Competencies

As demonstrated in the above section and in the submitted reflections, the extensive work that I have completed has helped me progress my learning in numerous ways. Throughout the project I have had a number of opportunities to engage with the team at Opengear beyond the academic scope of this course. These opportunities include a number of internal presentations, the company hackathon, and other internal company events. This has allowed me to develop my competencies to a more full extent. I believe that given this scope, I have had the opportunity to develop across all of the EA Stage 1 competencies.